IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

LIBERTY PATENTS, LLC, Plaintiff, v. MICROCHIP TECHNOLOGY INCORPORATED, NXP SEMICONDUCTORS N.V., NXP B.V., NXP USA, INC. D/B/A NXP SEMICONDUCTORS USA, INC., RENESAS ELECTRONICS CORPORATION, RENESAS ELECTRONICS AMERICA INC. D/B/A RENESAS AMERICA, STMICROELECTRONICS N.V., STMICROELECTRONICS INTERNATIONAL N.V., and STMICROELECTRONICS INC., Defendants.	CIVIL ACTION NO. 6:21-cv-78 ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT **JURY TRIAL DEMANDED**

ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Liberty Patents, LLC ("Liberty Patents" or "Plaintiff") files this original complaint against Defendants Microchip Technology Incorporated, NXP Semiconductors N.V., NXP B.V., NXP USA, Inc. d/b/a NXP Semiconductors USA, Inc., Renesas Electronics Corporation, Renesas Electronics America Inc. d/b/a Renesas America, STMicroelectronics N.V., STMicroelectronics International N.V., and STMicroelectronics Inc. (collectively "Defendants"), alleging, based on its own knowledge as to itself and its own actions and based on information and belief as to all other matters, as follows:

PARTIES

1. Liberty Patents is a limited liability company formed under the laws of the State of Texas, with its principal place of business at 2325 Oak Alley, Tyler, Texas, 75703.

2. Defendant Microchip Technology Incorporated ("Microchip") is a corporation organized and existing under the laws of Delaware. Microchip may be served with process through its registered agent, CT Corporation System, at 1999 Bryan St., Suite 900, Dallas, Texas, 75201.

3. Microchip describes itself as a provider of smart, connected and secure embedded control solutions.[1] It develops, manufactures, and sells specialized semiconductor products used by its customers for a wide variety of embedded control applications.[2] Its "solutions serve more than 120,000 customers across the industrial, automotive, consumer, aerospace and defense, communications and computing markets."[3]

4. Defendant NXP Semiconductors N.V. is a company organized and existing under the laws of The Netherlands. NXP Semiconductors N.V. has an office at High Tech Campus 60, 5656 AG Eindhoven, The Netherlands. NXP Semiconductors N.V. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas, 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

---

[1] *See* www.microchip.com/en-us/about/corporate-overview.

[2] *See* Microchip's Form 10-K, at 3 (May 21, 2020), www.microchip.com/content/dam/mchp/documents/financial/annual/fy20/pdf/Form%2010-K%20Filed%205-21-2020.pdf.

[3] *See* www.microchip.com/en-us/about/corporate-overview.

5. Defendant NXP B.V. is a company organized and existing under the laws of The Netherlands. NXP B.V. has an office at High Tech Campus 60, 5656 AG Eindhoven, The Netherlands. NXP B.V. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas, 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

6. NXP B.V. is a wholly owned subsidiary of NXP Semiconductors N.V. NXP B.V. develops and sells semiconductor devices globally. Its corporate parent, NXP Semiconductors N.V., does business globally through NXP B.V and NXP B.V.'s subsidiary companies.

7. Defendant NXP USA, Inc. d/b/a NXP Semiconductors USA, Inc. ("NXP USA") is a corporation organized and existing under the laws of Delaware. NXP USA may be served with process through its registered agent, Corporation Service Company d/b/a/ CSC-Lawyers Incorporating Service Company, at 211 East 7th Street, Suite 620, Austin, Texas, 78701-3218.

8. NXP USA is a subsidiary of both NXP Semiconductors N.V. and NXP B.V. According to its website, NXP operates three wafer fabrication facilities in the US through NXP USA—two of which are in Austin, Texas. These facilities manufacture "microcontrollers (MCUs) and microprocessors (MPUs), power management devices, RF transceivers, amplifiers and sensors."[4]

9. The Defendants identified in paragraphs 4 through 8 above (collectively, "NXP") are companies which together comprise "a global semiconductor company and a long-standing supplier in the industry, with over 50 years of innovation and operating history."[5] According to

[4] *See* www.nxp.com/company/about-nxp/worldwide-locations/united-states:USA.

[5] *See* NXP Semiconductors N.V.'s Form 10-K Annual Report, at 3 (Feb. 27, 2020), https://investors.nxp.com/sec-filings/sec-filing/10-k/0001413447-20-000009.

NXP, it provides technology solutions in the fields of cryptography-security, high-speed interface, radio frequency (RF), mixed-signal analog-digital (mixed A/D), power management, digital signal processing and embedded system design. Its products are used in a wide range of end-market applications, including automotive, industrial & Internet of Things (IoT), mobile, and communication infrastructure.[6]

10. The NXP Defendants named above and their affiliates are part of the same corporate structure and distribution chain for the making, importing, offering to sell, selling, and using of the accused devices in the United States, including in the State of Texas generally and this judicial district in particular. For example, NXP explains that the commercial name for NXP group of companies is "NXP" or "NXP Semiconductors."[7]

11. The NXP Defendants named above and their affiliates share the same management, common ownership, advertising platforms, facilities, distribution chains and platforms, and accused product lines and products involving related technologies.

12. Thus, the NXP Defendants named above and their affiliates operate as a unitary business venture and are jointly and severally liable for the acts of patent infringement alleged herein.

13. Renesas Electronics Corporation is a corporation organized under the laws of Japan. Renesas Electronics Corporation has an office at Toyosu Foresia, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan. Renesas Electronics Corporation may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas, 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a

[6] *Id.*

[7] *Id.*

resident agent for service of process in Texas as required by statute. This action arises out of that business.

14. Renesas Electronics America Inc. d/b/a Renesas America ("Renesas America") is a corporation organized and existing under the laws of California. Renesas America may be served with process through its registered agent, Corporation Service Company located at 251 Little Falls Drive, Wilmington, Delaware, 19808.

15. Renesas America is a wholly owned subsidiary of Renesas Electronics Corporation. It is a supplier of semiconductor products, including microcontrollers and SoCs.

16. The Defendants identified in paragraphs 13 through 15 above (collectively, "Renesas") are companies which together comprise one of the major suppliers of "advanced semiconductor solutions."[8] Renesas describes itself as a "leader in microcontrollers, analog, power, and SoC products" that "provides comprehensive solutions for a broad range of automotive, industrial, infrastructure, and loT applications."[9] It further describes itself as a company that "delivers trusted embedded design innovation with complete semiconductor solutions that enable billions of connected, intelligent devices to enhance the way people work and live."[10]

17. The Renesas Defendants named above and their affiliates are part of the same corporate structure and distribution chain for the making, importing, offering to sell, selling, and

---

[8] *See Renesas Electronics Announces Company Name Change of Consolidated Subsidiary* (Oct. 27, 2017), www.renesas.com/us/en/about/press-room/renesas-electronics-announces-company-name-change-consolidated-subsidiary.

[9] *Id*.

[10] *See Integrated Device Technology to Start Operations as Renesas Electronics America in January 2020 Following Successful Completion of Integration in US*. (Jan. 6, 2020), www.renesas.com/jp/en/about/press-room/integrated-device-technology-start-operations-renesas-electronics-america-january-2020-following.

using of the accused devices in the United States, including in the State of Texas generally and this judicial district in particular. Renesas has over 1,000 U.S. employees and has invested hundreds of millions of dollars in R&D in the United States.

18. The Renesas Defendants named above and their affiliates share the same management, common ownership, advertising platforms, facilities, distribution chains and platforms, and accused product lines and products involving related technologies.

19. Thus, the Renesas Defendants named above and their affiliates operate as a unitary business venture and are jointly and severally liable for the acts of patent infringement alleged herein.

20. STMicroelectronics N.V. is a company organized under the laws of The Netherlands. STMicroelectronics N.V. has an office at WTC Schiphol Airport, Schiphol Boulevard 265, 1118 BH Schiphol, The Netherlands. STMicroelectronics N.V. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas, 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

21. STMicroelectronics International N.V. is a company organized under the laws of The Netherlands. STMicroelectronics International N.V. has an office at WTC Schiphol Airport, Schiphol Boulevard 265, 1118 BH Schiphol, The Netherlands. STMicroelectronics International N.V. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas, 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

22. STMicroelectronics International N.V. is a wholly owned subsidiary of STMicroelectronics N.V. The Annual Report for the STMicroelectronics group of companies states that "[w]hile STMicroelectronics N.V. is the parent company, we conduct our global business through STMicroelectronics International N.V. and also conduct our operations through service activities from our subsidiaries."[11]

23. STMicroelectronics Inc. is a corporation organized and existing under the laws of Delaware. STMicroelectronics Inc. may be served with process through its registered agent, CT Corporation System, at 1999 Bryan St., Suite 900, Dallas, Texas, 75201.

24. STMicroelectronics Inc. is a subsidiary of STMicroelectronics N.V. STMicroelectronics Inc. provides manufacturing services for electronics products including semiconductors, multimedia products, power applications, and sensors.

25. The Defendants identified in paragraphs 20 through 24 above (collectively, "STMicroelectronics") are companies which together comprise a global independent semiconductor group that designs, develops, manufactures and markets a broad range of products, including discrete and standard commodity components, application-specific integrated circuits (ASICs), full custom devices and semi-custom devices and application-specific standard products (ASSPs) for analog, digital and mixed-signal applications.[12] STMicroelectronics states that its "operations are also conducted through [its] various subsidiaries, which are organized and

[11] *See* STMicroelectronics Annual Report (20-F) at 28 (2019), https://investors.st.com/static-files/122c173f-920f-44d3-bdb1-431a795b97f1.

[12] *See* STMicroelectronics Semi Annual Report at 23 (2020), https://investors.st.com/static-files/601d353d-aa59-46b3-ad3d-2009db640a12.

operated according to the laws of their country of incorporation, and consolidated by STMicroelectronics N.V."[13]

26. The STMicroelectronics Defendants named above and their affiliates are part of the same corporate structure and distribution chain for the making, importing, offering to sell, selling, and using of the accused devices in the United States, including in the State of Texas generally and this judicial district in particular.

27. The STMicroelectronics Defendants named above and their affiliates share the same management, common ownership, advertising platforms, facilities, distribution chains and platforms, and accused product lines and products involving related technologies.

28. Thus, the STMicroelectronics Defendants named above and their affiliates operate as a unitary business venture and are jointly and severally liable for the acts of patent infringement alleged herein.

29. The parties to this action are properly joined under 35 U.S.C. § 299 because the right to relief asserted against Defendants jointly and severally arises out of the same series of transactions or occurrences relating to the making and using of the same products or processes, including products certified by Platform Security Architecture (PSA) that automatically update the system software of an embedded system. Additionally, questions of fact common to all Defendants will arise in this action.

JURISDICTION AND VENUE

30. This is an action for infringement of a United States patent arising under 35 U.S.C. §§ 271, 281, and 284–85, among others. This Court has subject matter jurisdiction of the action under 28 U.S.C. § 1331 and § 1338(a).

[13] *See id*. at 18.

31. This Court has personal jurisdiction over Defendants pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) Defendants have done and continue to do business in Texas; (ii) Defendants have committed and continue to commit acts of patent infringement in the State of Texas, including making, using, offering to sell, and/or selling accused products in Texas, and/or importing accused products into Texas, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in Texas, and/or committing a least a portion of any other infringements alleged herein in Texas, and (iii) Defendants regularly place their products within the stream of commerce—directly, through subsidiaries, or through third parties—with the expectation and knowledge that such products will be shipped to, sold, or used in Texas and elsewhere in the United States. Thus, Defendants have established minimum contacts within Texas and purposefully availed themselves of the benefits of Texas, and the exercise of personal jurisdiction over Defendants would not offend traditional notions of fair play and substantial justice. In addition, or in the alternative, this Court has personal jurisdiction over the foreign Defendants pursuant to Federal Rule of Civil Procedure 4(k)(2).

32. Venue is proper in this district under 28 U.S.C. § 1400(b) because (i) Microchip has committed and continues to commit acts of patent infringement in this district, including making, using, offering to sell, and/or selling accused products in this district, and/or importing accused products into this district, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in this district, and/or committing at least a portion of any other infringements alleged herein in this district, (ii) Microchip is registered to do business in Texas, and (iii) Microchip has a regular and established

place of business in this district, including at least at 8601 Ranch Road 2222, Park Centre, Bldg. 3, Austin, Texas 78730:

Texas - Global Sales & Distribution

Microchip Office

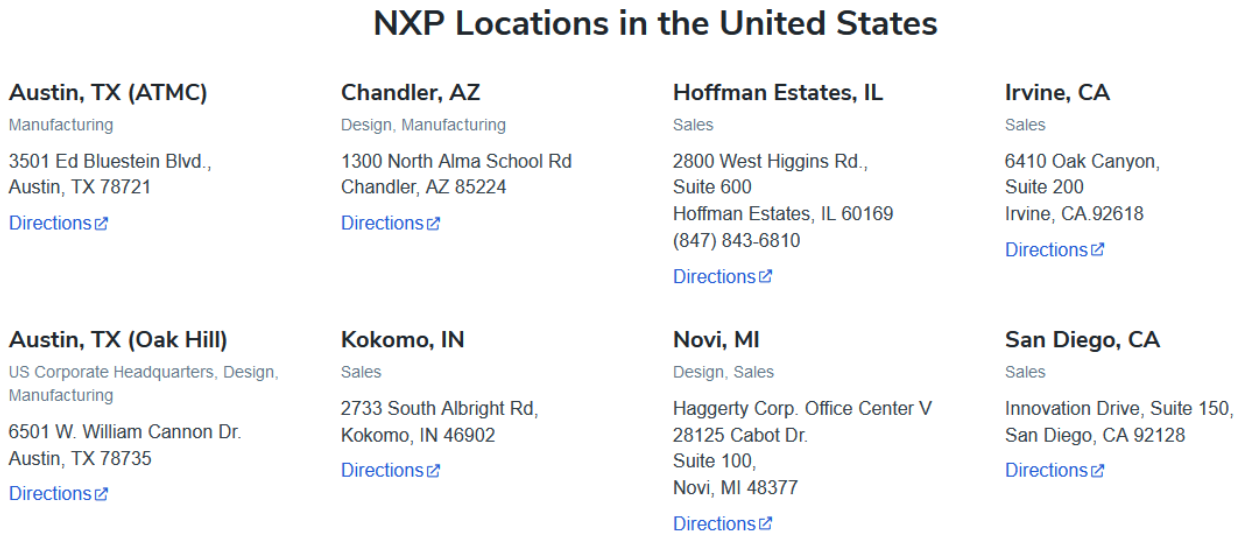
Microchip Technology Inc.
8601 Ranch Road 2222
Park Centre, Bldg. 3
Austin, Texas 78730
United States of America
Phone :(512) 502-0070
Fax :(512) 334-1984
Website : http://www.microchipdirect.com

Microchip Technology Inc.
2435 N. Central Expressway
Suite 460
Richardson, Texas 75080
United States of America
Phone :972-818-7423
Fax :972-818-2924
Website : http://www.microchipdirect.com

Microchip Technology Inc.
19500 State Highway 249
Suite 555
Houston, Texas 77070
United States of America
Phone :281 894-5983
Website : http://www.microchipdirect.com

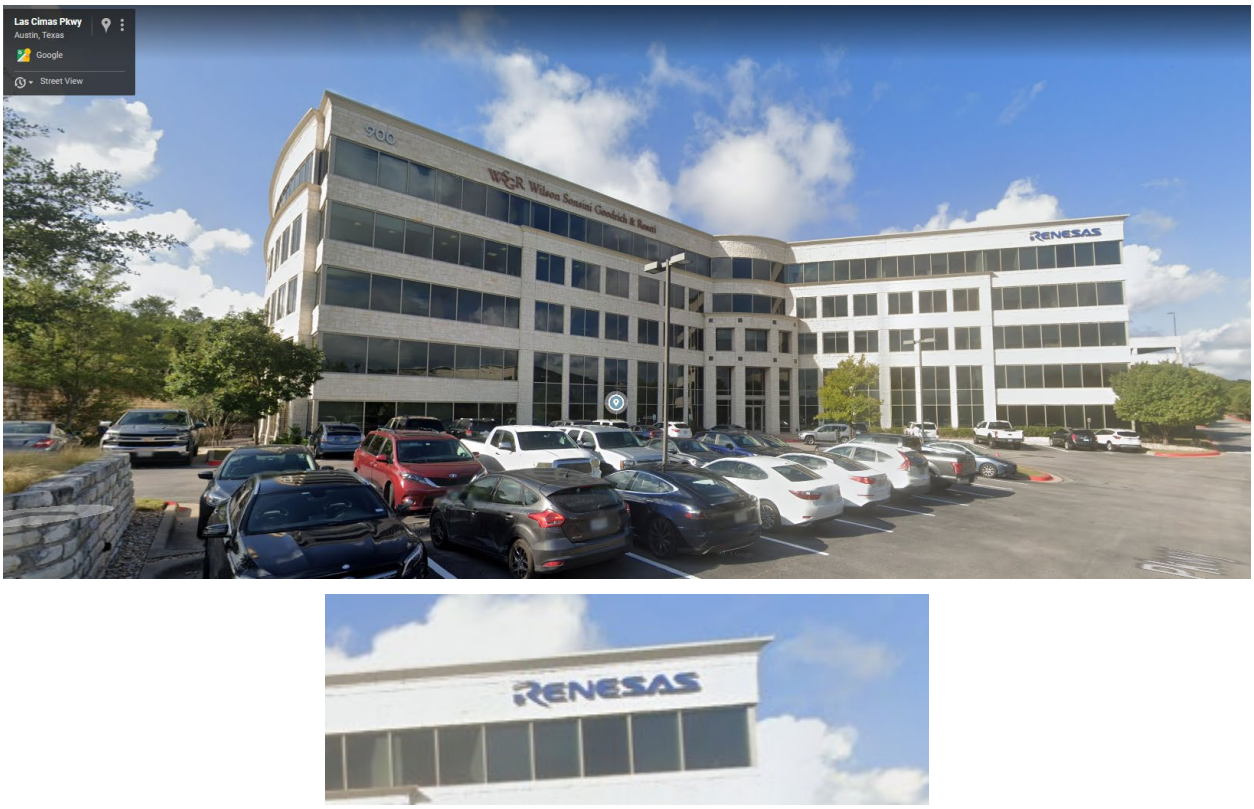
Source: www.microchip.com/salesdirectory/SalesListing/UNITED%20STATES/Texas

33. Venue is proper in this district under 28 U.S.C. § 1400(b) because (i) NXP has committed and continues to commit acts of patent infringement in this district, including making, using, offering to sell, and/or selling accused products in this district, and/or importing accused products into this district, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in this district, and/or committing at least a portion of any other infringements alleged herein in this district, (ii) NXP Semiconductors N.V. and NXP B.V. are foreign entities, (iii) NXP USA is registered to do business in Texas, and (iv) NXP USA has regular and established places of business in this district, including at least at 6501 W. William Cannon Drive, Austin, Texas, 78735, and at 3501 Ed Bluestein Blvd., Austin, Texas, 78721:



Source: www.nxp.com/company/about-nxp/worldwide-locations/united-states:USA

34. Venue is proper in this district under 28 U.S.C. § 1400(b) because (i) Renesas has committed and continues to commit acts of patent infringement in this district, including making, using, offering to sell, and/or selling accused products in this district, and/or importing accused products into this district, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in this district, and/or committing at least a portion of any other infringements alleged herein in this district, (ii) Renesas Electronics Corporation is a foreign entity; and (iii) Renesas America has a regular and established place of business in this district, including at least at 900 S. Capital of Texas Hwy, West Lake Hills, Texas, 78746:



Source: https://goo.gl/maps/7mB87SENEWE2aWKz5.

35. Venue is proper in this district under 28 U.S.C. § 1400(b) because (i) STMicroelectronics has committed and continues to commit acts of patent infringement in this district, including making, using, offering to sell, and/or selling accused products in this district, and/or importing accused products into this district, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in this district, and/or committing at least a portion of any other infringements alleged herein in this district, (ii) STMicroelectronics N.V. and STMicroelectronics International N.V. are foreign entities; (iii) STMicroelectronics Inc. is registered to do business in Texas, and (iv) STMicroelectronics Inc. has a regular and established place of business in this district, including at least at 8501 N. Mopac Expressway, #420, Austin, Texas, 78757:

First Alliance/Luscombe

2837 Mercy Drive
Fort Collins,80526, Colorado,United States of America
Phone:970-225-6498

FUTURE ELECTRONICS

30851 Agoura Rd, Suite 115 Agoura Hills
LOS ANGELES,95661, California,United States of America
Phone:+1 818 665 3957
Fax: +1 818 735 0764
Website

STMicroelectronics

8501 North Mo-Pac Expy Suite 420
AUSTIN,78759, Texas,United States of America
Phone:+1 512 225 6200

Source: www.st.com/content/st_com/en/contact-us.html

BACKGROUND

36. The patent-in-suit, U.S. Patent No. 7,493,612 ("the '612 Patent"), covers technology describing a better process for retrieving automatic software updates. Specifically, the '612 Patent discloses systems and methods for automatically updating the system software of an embedded system.

37. The invention of the '612 Patent relates to a method of automatically updating the system software of an embedded system using update agent interface programming (UAIP)—code that initiates an update of the system software during the boot process. The embedded system includes first system software and a boot image. The system also includes a micro-controller capable of transforming the first system software into system code and the boot image into boot code. The boot code includes update agent interface programming (UAIP) for initiating updating of the first system software before executing the system code. The system can be coupled to an external data storage device, which contains the second system software (i.e., the updated system code). If there is an update to the system software, the second system software is read from the external data storage device. As a result of the '612 Patent's inventive system, a computer can advantageously retrieve automatic updates during boot without loading its outdated OS—a more efficient, time-saving solution.

38. The '612 Patent's inventive system was developed by the Taiwanese company, Lite-On Technology Corp., which develops a wide range of consumer electronics products, such

as semiconductors, monitors, motherboards, etc. Lite-On was originally founded in 1975 by former employees of Texas Instruments. While the company originally developed LEDs, it branched into other industries, such as embedded systems and related software, and stayed on the forefront of developing technologies. Lite-On was recently purchased by the Japanese company, Kioxia—a former division of Toshiba—for $165 million.

39. The ’612 Patent discloses a novel and important invention that is highly relevant to today’s technology, which relies heavily on recurring updates to computer systems and IoT devices. It has been cited by major technology companies like Aruba Networks (now part of HPE), Google, IBM, Intelligent Platforms (now part of GE), Texas Instruments, and Vimicro.

COUNT I

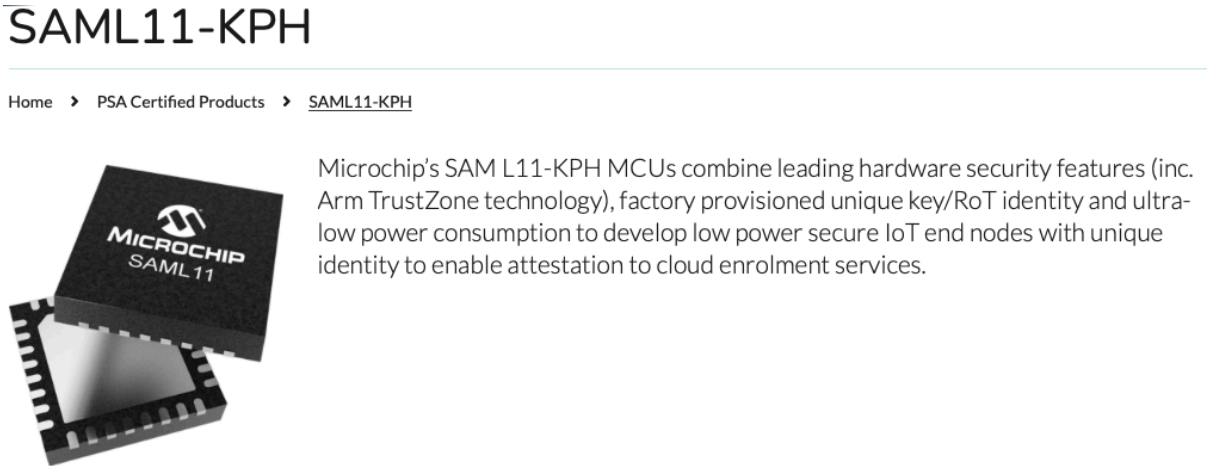
DIRECT INFRINGEMENT OF U.S. PATENT NO. 7,493,612

40. On February 17, 2009, the ’612 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Embedded System and Related Method Capable of Automatically Updating System Software.”

41. Liberty Patents is the owner of the ’612 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the ’612 Patent against infringers, and to collect damages for all relevant times.

42. Microchip made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, Microchip’s SAM L11-KPH MCUs and other products[14] that are certified by Platform Security Architecture (PSA) (“accused products”):

[14] *See, e.g.*, Microchip ATSAML11E14A-MU, ATSAML11E15A-MU, ATSAML11E14A-AUT, ATSAML11E15A-AFKPH, ATSAML11E16A-AFKPH, ATSAML11E16A-MUT, ATSAML11E16A-MFT, ATSAML11E16A-MFKPH, ATSAML11D15A-YU,



Source: www.psacertified.org/products/saml11-kph/

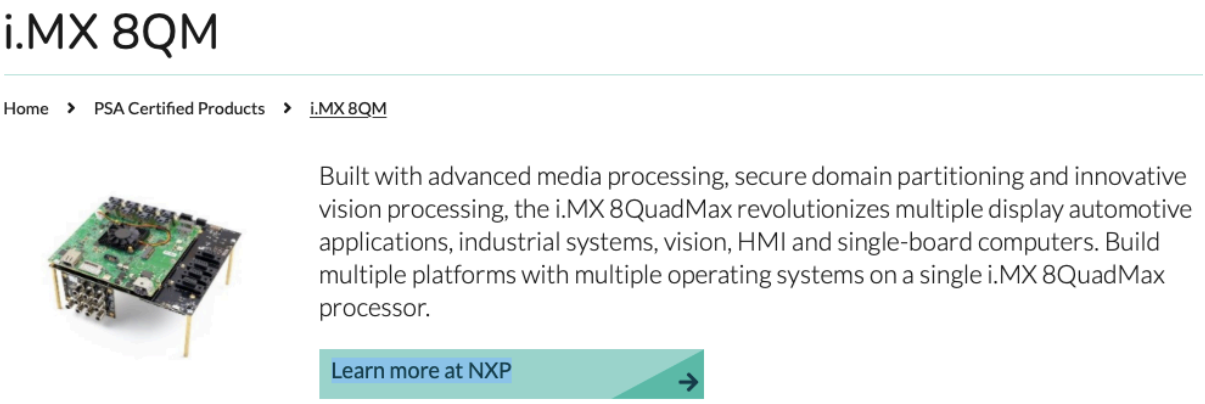
ATSAML11E16A-AUT, ATSAML11E16A-AU, ATSAML11D16A-MU, ATSAML11E16A-MU, ATSAML11E15A-MF, ATSAML11E15A-AU, ATSAML11D14A-YF, ATSAML11E16A-MF, ATSAML11E14A-AF, ATSAML11D16A-YF, ATSAML11E15A-AF, ATSAML11E15A-MUT, ATSAML11D14A-YUT, ATSAML11D15A-MU, ATSAML11E16A-AF, ATSAML11D15A-YF, ATSAML11E14A-AU, ATSAML11E16A-AUKPH, ATSAML11D16A-MF, ATSAML11D16A-MUT, ATSAML11E14A-MUT, ATSAML11E15A-MFKPH, ATSAML11E15A-AUKPH, ATSAML11E15A-AFT, ATSAML11E16A-AFTKPH, ATSAML11D16A-MFT, ATSAML11D15A-YUT, ATSAML11D14A-YFT, ATSAML11D16A-YFT, ATSAML11E15A-AUTKPH, ATSAML11D16A-MUTKPH, ATSAML11E15A-AUT, ATSAML11E15A-MFT, ATSAML11D16A-YU, ATSAML11D16A-YFKPH, ATSAML11D14A-MF, ATSAML11D16A-MUKPH, ATSAML11D16A-YUT, ATSAML11E16A-AUTKPH, ATSAML11E15A-MUTKPH, ATSAML11D14A-MUT, ATSAML11D15A-MUT, ATSAML11D14A-YFKPH, ATSAML11D16A-YUKPH, ATSAML11D15A-YFKPH, ATSAML11D14A-YUKPH, ATSAML11D14A-MU, ATSAML11D15A-YUKPH, ATSAML11D14A-YU, ATSAML11D14A-MUKPH, ATSAML11E14A-MUKPH, ATSAML11D15A-MUKPH, ATSAML11E14A-MF, ATSAML11E15A-MUKPH, ATSAML11D15A-MF, ATSAML11D14A-MFKPH, ATSAML11E14A-MFKPH, ATSAML11D15A-MFKPH, ATSAML11E16A-MUKPH, ATSAML11D16A-MFKPH, ATSAML11E14A-AUKPH, ATSAML11E14A-AFKPH, ATSAML11D14A-YUTKPH, ATSAML11D15A-YUTKPH, ATSAML11D15A-YFT, ATSAML11D14A-YFTKPH, ATSAML11D16A-YUTKPH, ATSAML11D15A-YFTKPH, ATSAML11D16A-YFTKPH, ATSAML11E14A-AUTKPH, ATSAML11E14A-AFT, ATSAML11E14A-AFTKPH, ATSAML11E15A-AFTKPH, ATSAML11E16A-AFT, ATSAML11E14A-MUTKPH, ATSAML11E14A-MFT, ATSAML11E14A-MFTKPH, ATSAML11E16A-MUTKPH, ATSAML11E15A-MFTKPH, ATSAML11D14A-MUTKPH, ATSAML11D14A-MFT, ATSAML11D15A-MUTKPH, ATSAML11E16A-MFTKPH, ATSAML11D15A-MFT, ATSAML11D14A-MFTKPH, ATSAML11D15A-MFTKPH, ATSAML11D16A-MFTKPH, DM320205), etc.

43. Microchip's SAM L11-KPH MCUs are exemplary accused products that are PSA Certified.

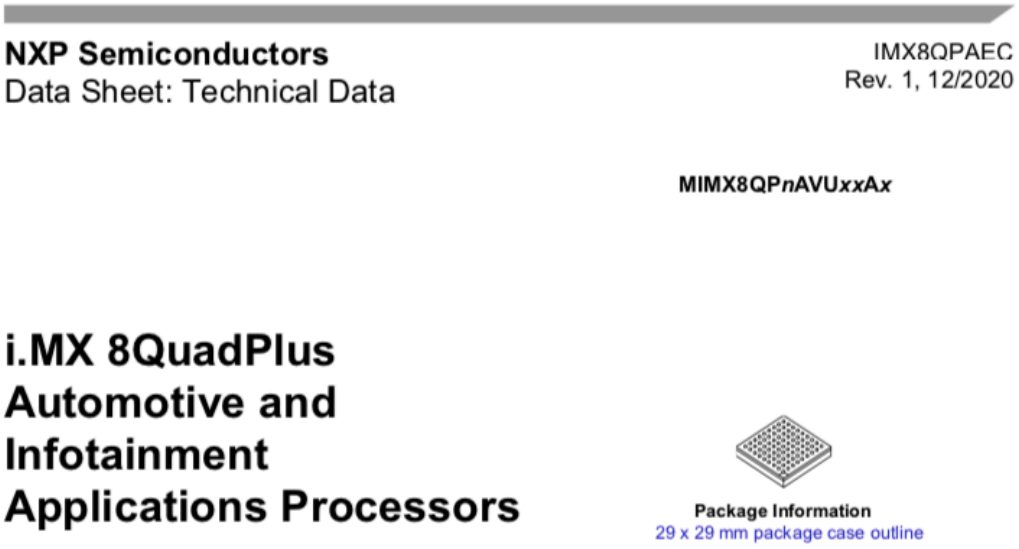
44. NXP made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, NXP's i.MX 8QM Applications Processors and other products[15] that are certified by Platform Security Architecture (PSA) ("accused products"):

---

[15] *See, e.g.*, NXP LPC55S16 (LPC55S16-EVK, LPC55S16JEV98E, LPC55S16JBD100E, LPC55S16JBD64E, LPC55S16JBD100K, LPC55S16JBD64K, LPC55S16JBD100Y, LPC55S16JEV98K, LPC55S16JBD64Y, LPC55S16JEV98Y); LPC55S00 (LPC55S06JBD64E, LPC55S06-EVK, LPC55S04JBD64E, LPC55S04JBD64K, LPC55S06JBD64K, LPC55S04JBD64Y, LPC55S06JBD64Y); i.MX 7ULP1 (MCIMX7U5DVP07SC, MCIMX7U5DVK07SC, MCIMX7U3DVK07SC, MCIMX7ULP-EVK, MCIMX7U5CVP06SC, MCIMX7U3CVP06SC, MCIMX7U5DVP08SC, MCIMX7U3CVP06SD, MCIMX7U5CVP06SD); i.MX 8M Mini (MIMX8MM2CVTKZAA, MIMX8MM5DVTLZAA, MIMX8MM4CVTKZAA, IMX8MM1DVTLZAA, MIMX8MM6DVTLZAA, MIMX8MM5CVTKZAA, MIMX8MM6CVTKZAA, MIMX8MM3DVTLZAA, MIMX8MM6DVTLZAAR, MIMX8MM4DVTLZAA, MIMX8MM3CVTKZAA, MIMX8MM1CVTKZAA, MIMX8MM2DVTLZAA); i.MX 8M Nano (MIMX8MN6DVTJZAA, MIMX8MN5DVTJZAA, MIMX8MN1DVTJZAA, MIMX8MN5CVTIZAA, MIMX8MN2CVTIZAA, MIMX8MN2DVTJZAA, MIMX8MN3DVTJZAA, MIMX8MN3CVTIZAA, MIMX8MN4DVTJZAA, MIMX8MN4CVTIZAA, MIMX8MN1CVTIZAA, MIMX8MN6CVTIZAA, MIMX8MN2CVTIZAAR); i.MX 8QM (MIMX8QM5AVUFFAB, MIMX8QM6AVUFFAB, MIMX8QM6CVUFFAB, MIMX8QM5CVUFFAB); i.MX 8QuadXPlus (MIMX8QX1AVLFZAC , MIMX8QX5AVLFZAC, MIMX8QX6AVLFZAC, MIMX8QX6CVLFZAC, MIMX8QX5CVLFZAC, MIMX8QX2AVLFZAC, MIMX8QX2AVOFZAC, MIMX8QX1AVOFZAC); i.MX RT600 (MIMXRT685-EVK, MIMXRT685SFVKB, MIMXRT685SFAWBR, MIMXRT633SFVKB, MIMXRT685SFFOB, MIMXRT685SFFOBR); i.MX RT1050 (MIMXRT1052CVL5A, MIMXRT1052CVL5B, MIMXRT1052DVJ6B, MIMXRT1051DVJ6B, MIMXRT1051CVJ5B, MIMXRT1052CVJ5B, MIMXRT1052DVL6B, MIMXRT1051CVL5B, MIMXRT1051DVL6B, MIMXRT1052CVL5BR, MIMXRT1051CVL5A, MIMXRT1051DVL6BR, MIMXRT1052DVL6BR, MIMXRT1051DVL6A); i.MX RT1060 (MIMXRT1061CVJ5B, MIMXRT106FDVL6A, MIMXRT106LDVL6A, MIMXRT1064-EVK, MIMXRT1062CVJ5A, MIMXRT1060-EVK, MIMXRT106ADVL6A, MIMXRT1064CVJ5A, MIMXRT1061CVL5A, MIMXRT1062CVL5A, MIMXRT1062DVJ6A, MIMXRT1064DVL6A, MIMXRT1064DVJ6A, IMXRT1064CVL5B, MIMXRT1062DVJ6B, MIMXRT1061DVL6B, MIMXRT1064DVL6B, MIMXRT1062DVL6B, IMXRT1062CVL5B, MIMXRT1064DVJ6B, MIMXRT1061CVL5B, MIMXRT106FDVL6B, MIMXRT106ADVL6B, MIMXRT1062CVJ5B,



Source: https://www.psacertified.org/products/i-mx-8qm/



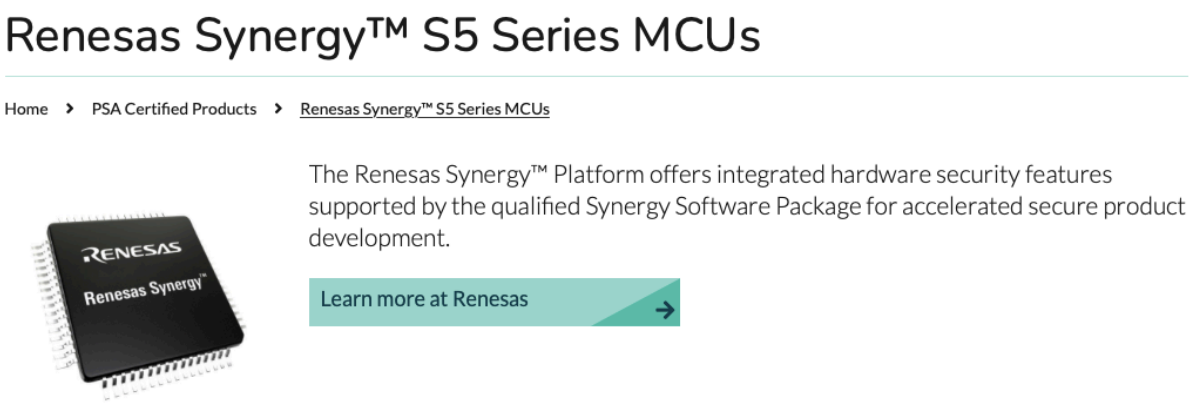
Source: https://www.nxp.com/docs/en/data-sheet/IMX8QPAEC.pdf

45. NXP's i.MX 8QM Applications Processors are exemplary accused products that are PSA Certified.

MIMXRT106FCVL5B, MIMXRT1061DVJ6A, MIMXRT1064CVJ5B, MIMXRT1061DVJ6B, MIMXRT1064DVL6AR, MIMXRT106CDVL6A, MIMXRT106CDVL6B, MIMXRT106SDVL6A, MIMXRT106SDVL6B, MIMXRT106ACVL5B, MIMXRT106LDVL6B, MIMXRT106LCVL5B, IMXRT1061CVJ5BR, MIMXRT1062CVJ5BR, MIMXRT1062CVJ5AR, MIMXRT106ADVL6BR, MIMXRT106ADVL6AR, MIMXRT1064DVL6BR); etc.

46. Reneseas made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, Renesas' Synergy™ S5 Series MCUs and other products[16] that are certified by Platform Security Architecture (PSA) ("accused products"):

[16] *See, e.g.*, Renesas RA4M3 (R7FA4M3AF3CFB#AA0, R7FA4M3AF3CFP#AA0, R7FA4M3AF3CFM#AA0, R7FA4M3AE3CFB#AA0, R7FA4M3AE3CFP#AA0, R7FA4M3AE3CFM#AA0, R7FA4M3AD3CFB#AA0); RA6M1 (R7FA6M1AD2CLJ#AC0, R7FA6M1AD3CFM#AA0, R7FA6M1AD3CFP#AA0, R7FA6M1AD3CNB#AC0); RA6M2 (R7FA6M2AD2CLK#AC0, R7FA6M2AD3CFB#AA0, R7FA6M2AD3CFP#AA0, R7FA6M2AF2CLK#AC0, R7FA6M2AF3CFB#AA0, R7FA6M2AF3CFP#AA0); RA6M3 (R7FA6M3AF3CFP#AA0, R7FA6M3AF2CBG#AC0, R7FA6M3AH2CBG#AC0, R7FA6M3AH3CFC#AA0, R7FA6M3AH3CFP#AA0, R7FA6M3AF3CFC#AA0, R7FA6M3AH2CLK#AC0, R7FA6M3AF3CFB#AA0, R7FA6M3AH3CFB#AA0, R7FA6M3AF2CLK#AC0); RA6M4 (R7FA6M4AF3CFP#AA0, R7FA6M4AD3CFB#AA0, R7FA6M4AD3CFM#AA0, R7FA6M4AD3CFP#AA0, R7FA6M4AE3CFB#AA0, R7FA6M4AE3CFM#AA0, R7FA6M4AE3CFP#AA0, R7FA6M4AF3CFB#AA0, R7FA6M4AF3CFM#AA0); RA6T1 (R7FA6T1AB3CFP#AA0, R7FA6T1AB3CFM#AA0, R7FA6T1AD3CFM#AA0, R7FA6T1AD3CFP#AA0); S5D3 (R7FS5D37A3A01CFP#AA0, R7FS5D37A3A01CFM#AA0 , R7FS5D37A3A01CNB#AC0, R7FS5D37A2A01CLJ#AC0, R7FS5D37A3A01CFM#HA0, R7FS5D37A3A01CFP#HA0, R7FS5D37A3A01CFM#BA0, R7FS5D37A3A01CFP#BA0, R7FS5D37A3A01CNB#HC0); S5D5 (R7FS5D57A3A01CFB#AA0, R7FS5D57A3A01CFP#AA0, R7FS5D57C3A01CFP#AA0, R7FS5D57A2A01CLK#AC0, R7FS5D57C2A01CLK#AC0, R7FS5D57C3A01CFB#AA0, R7FS5D57A2A01CLK#AC1, R7FS5D57A3A01CFB#AA1, R7FS5D57A3A01CFP#AA1, R7FS5D57C2A01CLK#AC1, R7FS5D57C3A01CFB#AA1, R7FS5D57C3A01CFP#AA1, R7FS5D57A3A01CFB#BA1, R7FS5D57A3A01CFP#BA1, R7FS5D57C3A01CFB#BA1, R7FS5D57C3A01CFP#BA1); S5D9 (R7FS5D97C3A01CFP#AA0, R7FS5D97C3A01CFB#AA0, R7FS5D97C2A01CLK#AC0, R7FS5D97E3A01CFP#AA0, R7FS5D97E3A01CFB#AA0, R7FS5D97E2A01CLK#AC0, R7FS5D97E3A01CFC#AA0, R7FS5D97E2A01CBG#AC0, R7FS5D97C2A01CBG#AC0, R7FS5D97C3A01CFC#AA0, R7FS5D97C3A01CFP#BA0, R7FS5D97C3A01CFB#BA0, R7FS5D97E3A01CFB#BA0, R7FS5D97E3A01CFC#BA0, R7FS5D97E3A01CFP#BA0, R7FS5D97C3A01CFC#BA0); etc.



Source: www.psacertified.org/products/renesas-synergy-s5-series-mcus/

47. Renesas' Synergy™ S5 Series MCUs are exemplary accused products that are PSA Certified.

48. STMicroelectronics made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, STMicroelectronics' STM32L5 series of Ultra-low-power MCUs and othe products[17] that are certified by Platform Security Architecture (PSA) ("accused products"):

[17] *See, e.g.*, STMicroelectronics STM32L4 (STM32L431RC, STM32L433VC, STM32L496ZE, STM32L412XX, STM32L422XX, STM32L431XX, STM32L432XX, STM32L433XX, STM32L442KC, STM32L443XX, STM32L451XX, STM32L452XX, STM32L462XX, STM32L471XX, STM32L475XX, STM32L476XX, STM32L486XX, STM32L496XX, STM32L4A6AG, STM32L4A6QG, STM32L4A6RG, STM32L4A6VG, STM32L4A6ZG, STM32L412T8, STM32L412XXXX, STM32L412XXXXP, STM32L412XXXXTR, STM32L412XXXXPTR, STM32L422XXXX, STM32L422XXXXP, STM32L422XXXXTR, STM32L431XXXX, STM32L431XXXXTR, STM32L432XXXX, STM32L432XXXXTR, STM32L433XXXX, STM32L433XXXXP, STM32L433XXXXTR, STM32L442XXXX, STM32L442XXXXTR, STM32L443XXXX, STM32L443XXXXTR, STM32L451XXXX, STM32L451XXXXTR, STM32L452XXXX, STM32L452XXXXP, STM32L452XXXXTR, STM32L462XXXX, STM32L462XXXXTR, STM32L471XXXX, STM32L471XXXXTR, STM32L475XXXX, STM32L475XXXXTR, STM32L476XXXX, STM32L476XXXXP, STM32L476XXXXTR, STM32L476XXXXPTR, STM32L476XXXXVTR, TM32L476XXXXMTR, STM32L476G-EVAL, STM32L486XXXX, STM32L486XXXXTR, STM32L496G-DISCO, STM32L496XXXX, STM32L496XXXXP, STM32L496XXXXTR, STM32L496XXXXPTR, STM32L4A6XXXX, STM32L4A6XXXXP, STM32L4A6XXXXTR, STM32L4A6XXXXPTR, STM32L4P5XXXX, STM32L4P5XXXXP, STM32L4P5XXXXTR, STM32L4P5XXXXPTR, STM32L4Q5XXXX, STM32L4Q5XXXXP, STM32L4Q5XXXXTR,

STM32L4Q5XXXXPTR, STM32L4R5XXXX, STM32L4R5XXXXP, STM32L4R5XXXXS, STM32L4R5XXXXTR, STM32L4R5XXXXSTR, STM32L4R7XXXX, STM32L4R9XXXX, STM32L4R9XXXXTR, STM32L4R9XXXXPTR, STM32L4R9I-DISCO, STM32L4R9I-EVAL, STM32L4S5XXXX, STM32L4S5XXXXP, STM32L4S5XXXXTR, STM32L4S7XXXX, STM32L4S9XXXX, STM32L4S9XXXXTR); STM32L5 (STM32L562RE, STM32L552CC, STM32L552CE, STM32L552ME, STM32L552QC, STM32L552QE, STM32L552RC, STM32L552RE, STM32L552VC, STM32L552VE, STM32L552ZC, STM32L552ZE, STM32L562CE, STM32L562ME, STM32L562QE, STM32L562VE, STM32L562ZE, STM32L552CCU6, STM32L552CEU6, STM32L562CEU6, STM32L552CET6, STM32L562VET6, STM32L552ZET6Q, STM32L562E-DK, STM32L552CCT6, STM32L562RET6, STM32L552VET6, STM32L562CET6, STM32L552RCT6, STM32L552E-EV, STM32L552ZET6, STM32L562ZET6, STM32L552CEU6P, STM32L562CEU6P, STM32L552RET6Q, STM32L552ZCT6Q, STM32L562QEI6Q, STM32L562QEI6, STM32L552QEI6, STM32L552VET6Q, STM32L562MEY6PTR, STM32L562RET6P, STM32L562VET6Q, STM32L552RET6, STM32L562CET6P, STM32L552QEI6Q, STM32L562ZET6Q, STM32L562QEI6TR, STM32L552CET6P, STM32L552MEY6PTR, STM32L552QEI6P, STM32L552MEY6QTR, STM32L552VCT6Q, STM32L552RET6P, STM32L562RET6Q, STM32L562VET6TR, STM32L562CET3P, STM32L562CEU6TR, STM32L552QCI6Q, STM32L562MEY6QTR, STM32L552CET6Q); etc.

Source: www.psacertified.org/products/stm32l5/; www.st.com/content/st_com/en/about/media-center/press-item.html/p4087.html; www.st.com/en/microcontrollers-microprocessors/stm32l5-series.html.

49. STMicroelectronics' STM32L5 series of Ultra-low-power MCUs are exemplary accused products that are PSA Certified.

50. By doing so, Defendants have directly infringed (literally and/or under the doctrine of equivalents) at least Claim 1 of the '612 Patent. Defendants' infringement in this regard is ongoing.

51. For example, the PSA Certified accused products are embedded systems capable of automatically updating system software. PSA (Platform Security Architecture) is a framework designed for providing security features for products such as ICs, chipsets, SOCs, etc. For a product to become PSA Certified, it must comply with specific hardware and software requirements. The accused products include an SoC. One feature of the device is the secure update process for updating the firmware ("system software") of the SoC ("embedded system").

52. The accused products include two sections of firmware: an Immutable section and an Updateable section. The Updateable section of the firmware receives updates over a network and is automatically updated without any physical intervention.

1 PSA overview

The Platform Security Architecture (PSA) is a framework for securing devices. Though the focus is on local network or internet connected devices, many aspects are relevant for non-connected devices. PSA includes a holistic set of deliverables, including Threat Models and Security Analyses documentation, hardware and firmware architecture specifications, APIs, and an API test suite. Together with an open source reference implementation, PSA enables consistent design-in at the right level of security. PSA-compliant products may go through a security evaluation and become PSA Certified™.

PSA builds upon best practice from across the industry and is aimed at different entities throughout the supply chain, from chip designers and device developers to cloud and network infrastructure providers and software vendors.

This document defines the overall security elements for designing and deploying trusted PSA-compliant devices within ecosystems. This document is the top-level document for the other PSA specifications and defines common language, high-level robustness rules, and models.

PSA is anchored in a set of core security goals, which provide a high-level, abstract, way to think about the essential features that are necessary to secure and establish trust. Abstraction allows these goals to be applied as required, for example, to an end user connected device, a hardware component, a software component, or a service. In describing the goals, the term *device* is used to represent any entity that must be secure and trustworthy.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 1).

Goal 5: Devices support secure update.

It is essential to update device software, or hardware configuration, to resolve security issues or to provide feature updates, without compromising device security. Authentication of an update is required. However, execution of any updated software must be authorized in accordance with Goal 4.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 2).

For the purpose of this document, the following terms are used to describe a generic PSA-compliant product:

Component	Description	Notes
Device	Final end product.	For example, a networked security camera or a tracking device for asset management.
System	Inseparable component integrating all processing elements, bus masters, and PSA Immutable Root of Trust.	Typically an SoC or equivalent. But could also include, for example, an external SIM or TPM device which is inseparably bound to the rest of the system by cryptographic or physical means.

Source:

https://armkeil.blob.core.windows.net/developer/Files/pdf/PlatformSecurityArchitecture/Architect/DEN0079-PSA_SM_ALPHA-02.pdf (Page 18).

4 PSA-RoT secure boot and firmware update

All PSA devices must support a secure boot flow to ensure only authorized software can be executed on the device. Secure boot, sometimes called verified boot, uses cryptography to verify the next stage code and any metadata. Execution of the next stage proceeds only if any validation checks on the verified metadata pass, for example, version comparison, see section 4.3. In some contexts, the term Measured Boot is used, however, this involves measuring the code, typically for attestation, but without any verification and validity checks.

The secure boot flow must start with an immutable and inherently trusted Boot ROM because it is the trust anchor for the boot validation chain. It is recommended, as shown in Figure 6, that:

- The Boot ROM is small, simple, and verifiable. This minimizes the risk of a vulnerability that cannot be corrected once on the chip, and
- The complex steps are handled by a main bootloader, subject to validity check by the Boot ROM, which can be corrected through a secure firmware update process, see section 4.5.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 14).

1.2 PSA device model

The PSA Device Model, shown in Figure 2, is a generic reference for defining security properties throughout this document. Designed and deployed devices should map to this reference model.

Figure 2 shows the following domains:

- The PSA Root of Trust (PSA-RoT), which is discussed in section 2, consists of:
  - The PSA Immutable Root of Trust, which is inherently trusted[1] and never changes on a production device.
  - The PSA Updateable Root of Trust, which is trusted only through verification that is anchored to the Immutable PSA Root of Trust. See section 4.
  - Trusted subsystems are, for example, any protected external memory, trusted peripherals, or a secure element, for example a SIM or TPM. Configuration and access control are protected by and attestable by the PSA-RoT.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 4).

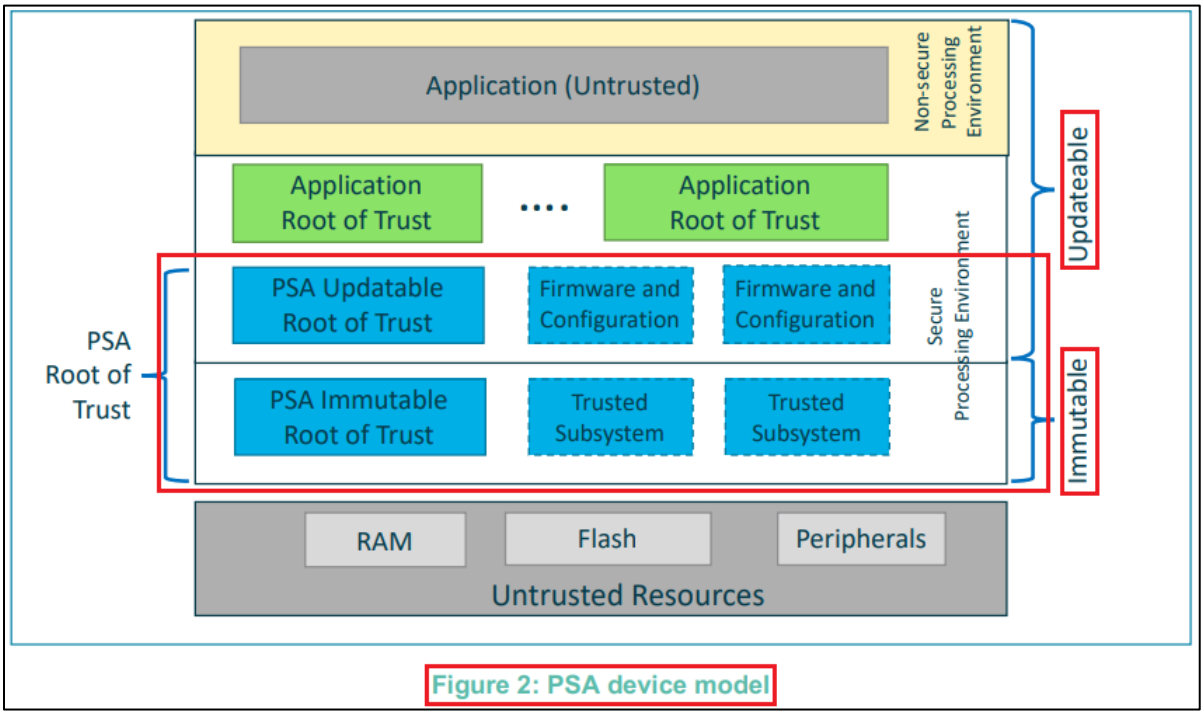


Figure 2: PSA device model

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).

This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).

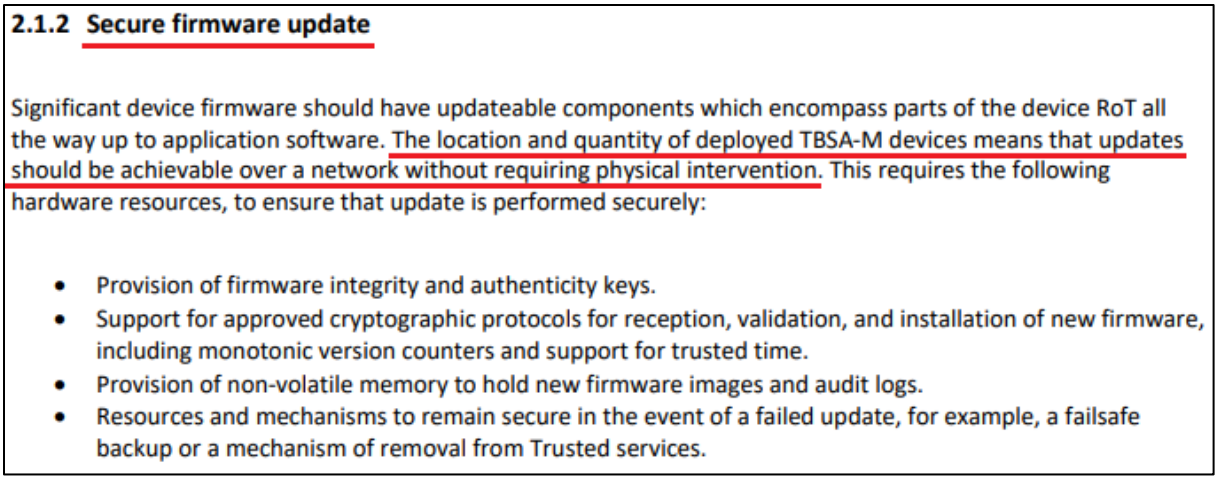
3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).



2.1.2 Secure firmware update

Significant device firmware should have updateable components which encompass parts of the device RoT all the way up to application software. The location and quantity of deployed TBSA-M devices means that updates should be achievable over a network without requiring physical intervention. This requires the following hardware resources, to ensure that update is performed securely:

- Provision of firmware integrity and authenticity keys.
- Support for approved cryptographic protocols for reception, validation, and installation of new firmware, including monotonic version counters and support for trusted time.
- Provision of non-volatile memory to hold new firmware images and audit logs.
- Resources and mechanisms to remain secure in the event of a failed update, for example, a failsafe backup or a mechanism of removal from Trusted services.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

53. The accused products include a first storage device for storing a first system software and a boot image.

54. For example, the accused products include two sections of firmware: an Immutable section and an Updateable section. The Immutable section of firmware contains code that is executed immediately after powering on the system. The Immutable section ("boot image") and the Updateable section ("first system software") of firmware are stored in the embedded flash ("first storage device").

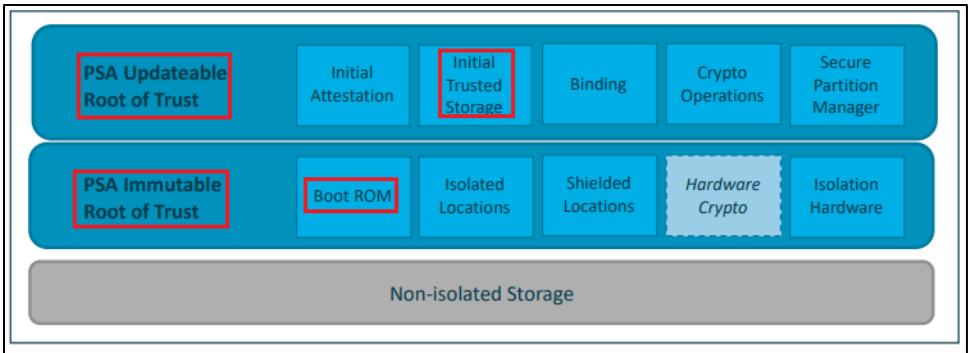


Figure 3: Minimal set of PSA services

2.1 Immutable elements

The PSA Immutable Root of Trust includes the following elements:

- The Boot ROM, which contains the first code to execute after release from reset. This means that the Boot ROM is the ultimate root of trust for all software that follows. The Boot ROM must be on-chip, and is typically implemented as Mask ROM, locked OTP, or locked on-chip flash. See section 4.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 6).

Non-Volatile Memory

The system on chip integrates several partitions of embedded Flash – the partitions are lockable by fuse enabling their contents to become immutable. Boot ROM is implemented in this manner. The ROM is written at manufacture and a permanent fuse is set so that subsequent update is not possible.

In addition there is several kilobits of One-Time-Programmable (OTP) efuses. These are used to store IDs, device keys and other secrets, and non-volatile device flags.

Source:

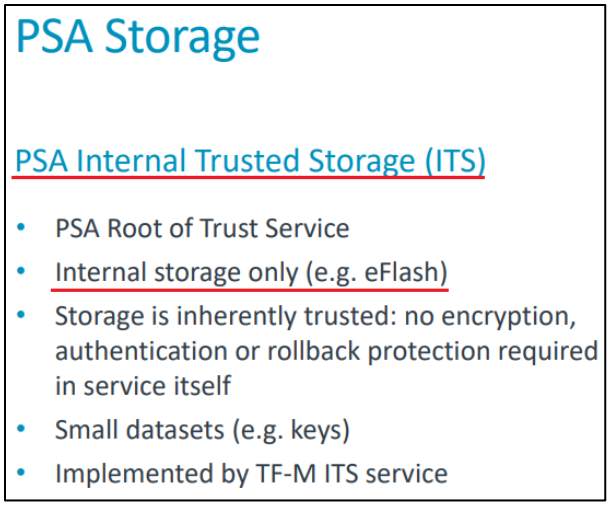
https://armkeil.blob.core.windows.net/developer/Files/pdf/PlatformSecurityArchitecture/Architect/DEN0079-PSA_SM_ALPHA-02.pdf (Page 70).

2.2 Updateable elements

The PSA Updateable Root of Trust includes the following elements:

- Secure Partition Management (SPM), which enforces partition level isolation-based access control policies. See section 2.3.
- Internal Trusted Storage (ITS), which provides secure partition-based access control to Isolated Locations and Shielded Locations. See section 5.1.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 7).



Source: https://www.trustedfirmware.org/docs/tfm_secure_storage_tech_forum.pdf (Page 3).

eFlash	See Internal flash
eFuse	OTP memory, available in very limited quantity
eMMC	Embedded multi media card. Low cost flash memory with a built-in controller
HMAC	Hashed Message Authentication Code
HUK	Hardware Unique Key
Internal flash	On-chip embedded flash

Source: https://pages.arm.com/rs/312-SAX-488/images/DEN0072-PSA_TBFU_1.0-bet1.pdf (Page 9).

55. The accused products include a micro-controller that is coupled to the first storage device for respectively transforming the first system software and the boot image into a system code and a boot code. The micro-controller orderly executes the boot code and the system code to control booting of the embedded system.

56. For example, the accused devices support Trusted Base System Architectures like TBSA-M and TBSA-A. TBSA-M specifies the architecture for products including certain ARM-based micro-controllers. The embedded flash of the accused products' micro-controller includes two sections of firmware images: an Immutable boot image ("boot image") and an Updateable firmware image ("first system software").

57. Before booting the system, a small program called boot loader (i.e., start-up code) uses linker files to map sections of the Immutable and Updateable firmware images to different parts of the memory for execution. That is, the firmware images are stored in one memory location before loading and in another memory location during execution. This is accomplished through ARM's linker files, which provide information about the mapping of different sections of the firmware images to different sections of memory. Once the mapping into memory is complete, the micro-controller's processor is then capable of executing the mapped Immutable firmware ("boot code") and the mapped Updateable firmware ("system code"). Accordingly, the micro-controller's processor transforms the "boot image" and the "system software" into memory-mapped executables (i.e., boot code and system code, respectively).

> This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:
>
> - *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
> - *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Pages 13 and 14).

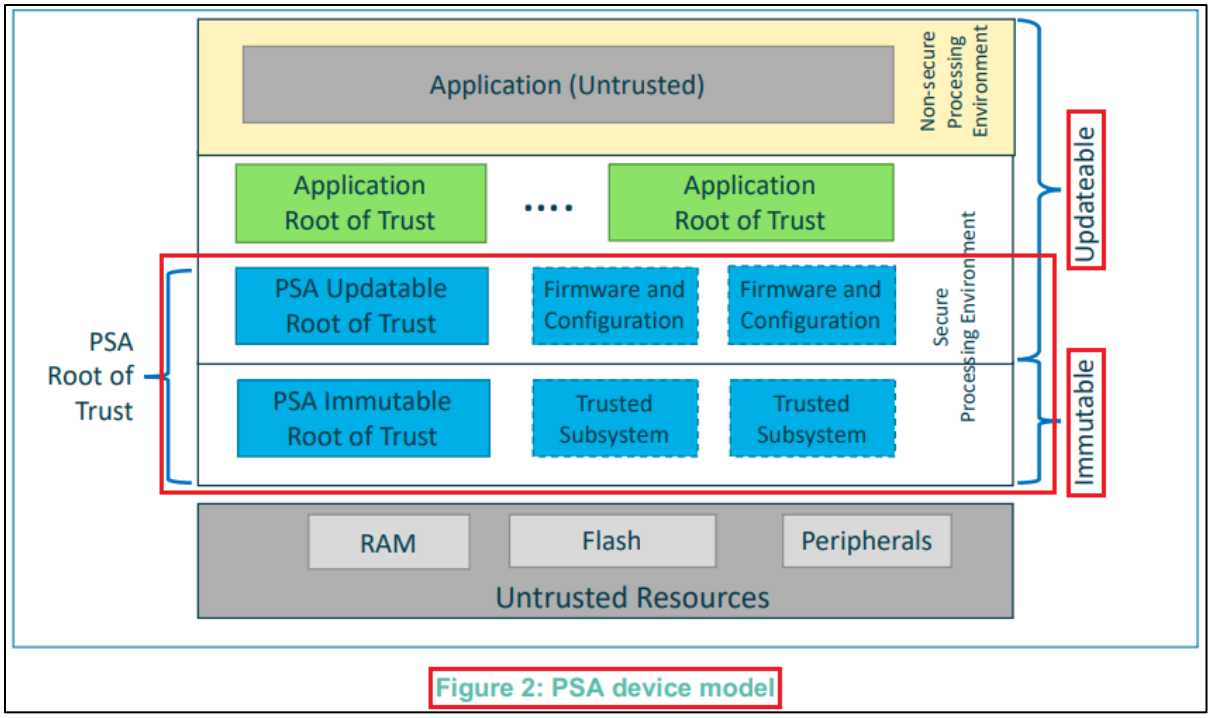


Figure 2: PSA device model

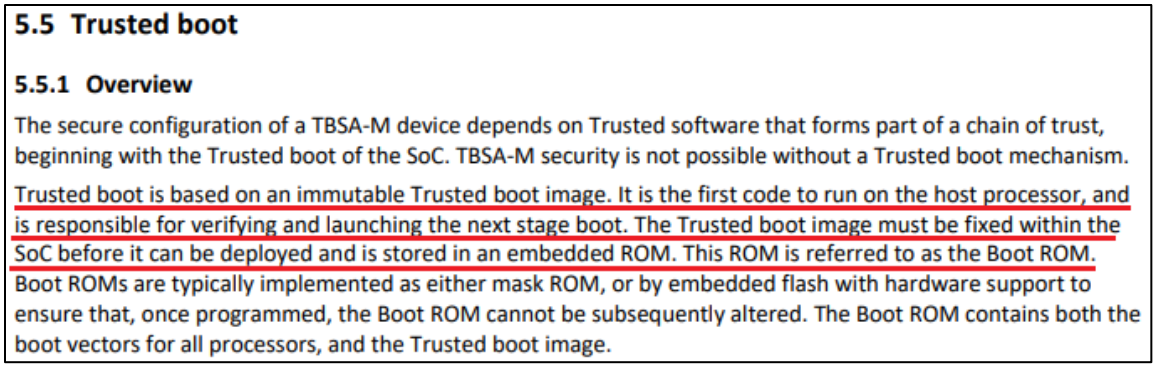
Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).

2.1 Immutable elements

The PSA Immutable Root of Trust includes the following elements:

- The Boot ROM, which contains the first code to execute after release from reset. This means that the Boot ROM is the ultimate root of trust for all software that follows. The Boot ROM must be on-chip, and is typically implemented as Mask ROM, locked OTP, or locked on-chip flash. See section 4.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 6).



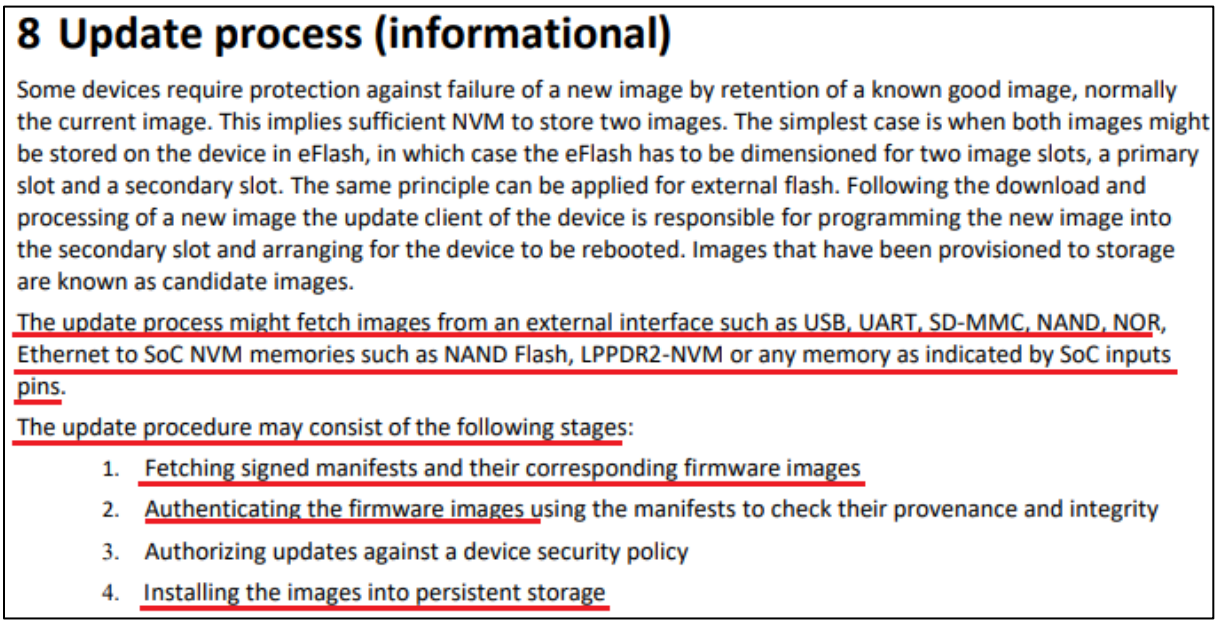
5.5 Trusted boot

5.5.1 Overview

The secure configuration of a TBSA-M device depends on Trusted software that forms part of a chain of trust, beginning with the Trusted boot of the SoC. TBSA-M security is not possible without a Trusted boot mechanism.

Trusted boot is based on an immutable Trusted boot image. It is the first code to run on the host processor, and is responsible for verifying and launching the next stage boot. The Trusted boot image must be fixed within the SoC before it can be deployed and is stored in an embedded ROM. This ROM is referred to as the Boot ROM. Boot ROMs are typically implemented as either mask ROM, or by embedded flash with hardware support to ensure that, once programmed, the Boot ROM cannot be subsequently altered. The Boot ROM contains both the boot vectors for all processors, and the Trusted boot image.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 50).



8 Update process (informational)

Some devices require protection against failure of a new image by retention of a known good image, normally the current image. This implies sufficient NVM to store two images. The simplest case is when both images might be stored on the device in eFlash, in which case the eFlash has to be dimensioned for two image slots, a primary slot and a secondary slot. The same principle can be applied for external flash. Following the download and processing of a new image the update client of the device is responsible for programming the new image into the secondary slot and arranging for the device to be rebooted. Images that have been provisioned to storage are known as candidate images.

The update process might fetch images from an external interface such as USB, UART, SD-MMC, NAND, NOR, Ethernet to SoC NVM memories such as NAND Flash, LPPDR2-NVM or any memory as indicated by SoC inputs pins.

The update procedure may consist of the following stages:

1. Fetching signed manifests and their corresponding firmware images
2. Authenticating the firmware images using the manifests to check their provenance and integrity
3. Authorizing updates against a device security policy
4. Installing the images into persistent storage

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 42).

2.4 Image

An image contains one or more of the following artifacts:

- Software executables
- Firmware executables
- Patches
- Configuration data and parameters

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 17)

4.1 The image structure

The structure of an image is defined by the:

- number of its constituent regions and output sections
- positions in memory of these regions and sections when the image is loaded
- positions in memory of these regions and sections when the image executes.

Each link stage has a different view of the image:

ELF object file view (linker input)

The ELF object file view comprises input sections. The ELF object file can be:

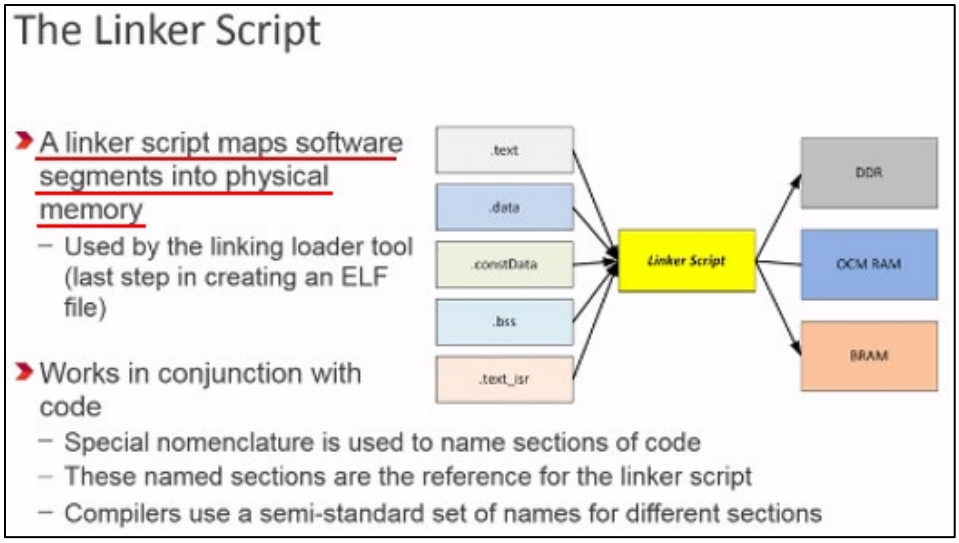
- a relocatable file that holds code and data suitable for linking with other object files to create an executable or a shared object file
- an executable file that holds a program suitable for execution
- a shared object file that holds code and data in the following contexts:
  - — the linker processes the file with other relocatable and shared object files to create another object file
  - — the dynamic linker combines the file with an executable file and other shared objects to create a process image.

Linker view The linker has two views for the address space of a program that become distinct in the presence of overlaid, position-independent, and relocatable program fragments (code or data):

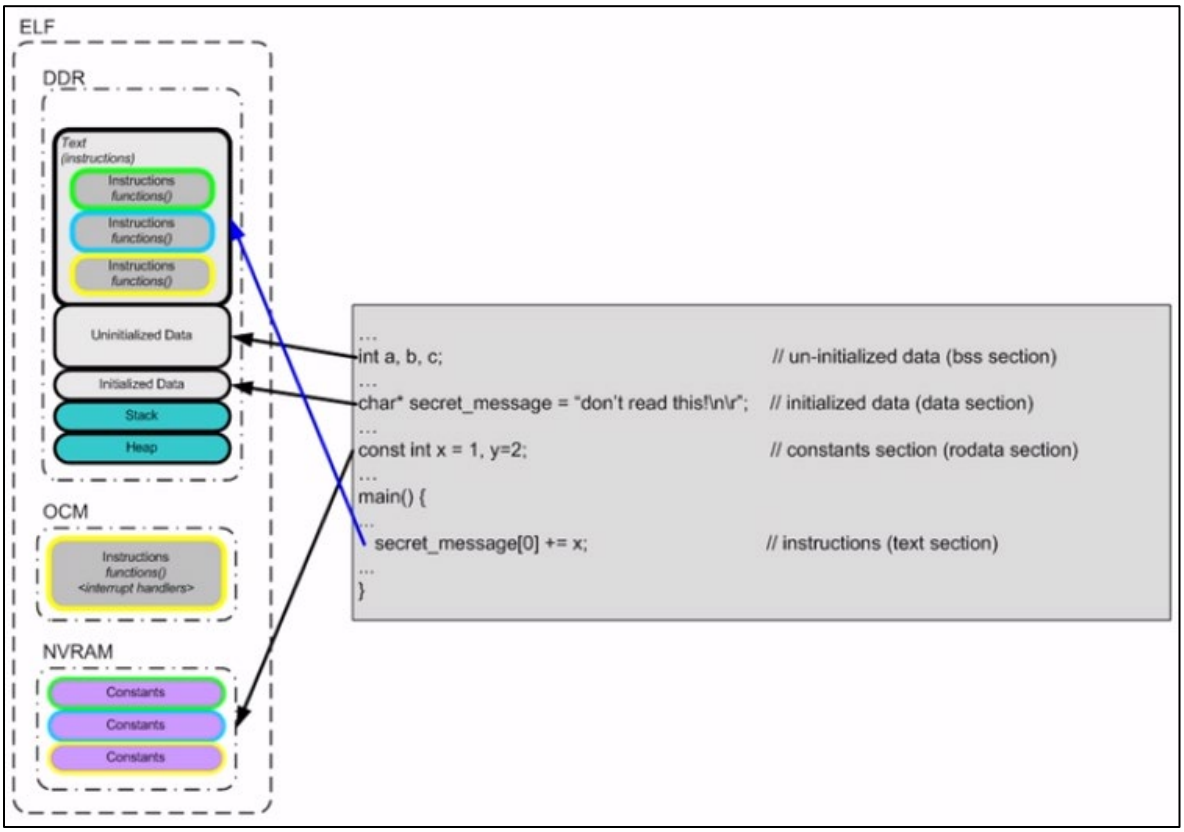
- The load address of a program fragment is the target address that the linker expects an external agent such as a program loader, dynamic linker, or debugger to copy the fragment from the ELF file. This might not be the address at which the fragment executes.
- The execution address of a program fragment is the target address where the linker expects the fragment to reside whenever it participates in the execution of the program.

If a fragment is position-independent or relocatable, its execution address can vary during execution.

Source: https://documentation-service.arm.com/static/5ea19a939931941038de91a1?token= (Page 32).



Source: https://www.xilinx.com/training/customer-training/using-linker-scripts.html (2:20)



Source: https://www.xilinx.com/training/customer-training/using-linker-scripts.html (9:52)

> The microcontroller boot process starts by simply applying power to the system. Once the voltage rails stabilize, the microcontroller looks to the reset vector for the location in flash where the start-up instruction can be found. The reset vector is a special location within the flash memory

Source: https://www.beningo.com/understanding-the-microcontroller-boot-process/

> The address that is stored at the reset vector is loaded by the microcontroller and the instructions that are contained there are then loaded and executed by the CPU. Now these first instructions aren't the start of main that the developer created. Instead, these are instructions on how to start-up the microcontroller.
>
> The first thing that usually occurs is that the vector tables that are stored in flash are copied to RAM. They are copied from and to the location that is specified in the linker file at the time the executable program is created. One reason for copying the vector tables to RAM is that it is faster to execute from RAM than flash. This helps to decrease the latency of any interrupt calls within the system. Depending on the particular architecture of the microcontroller there may then be an instruction to update a vector table register so that the microcontroller knows where the start of the RAM table is.
>
> Next the initialized data sections are copied into RAM. This is usually variables that are stored in the .data section of the linker. Examples of initialized data would be static, global and static local variables that have been provided with an initialization value during compile time. These are explicit definitions such as int Var = 0x32;.
>
> Following the copy of the data section, the .bss section is also copied. The .bss section contains variables that are not initialized explicitly or that have been initialized to a value of zero. A simple example is that the variable static int Var; would be contained within this section.
>
> Finally, the microcontroller will copy any RAM functions from flash to RAM. Once again it is sometimes worthwhile to execute certain functions out of RAM rather than flash due to the execution speed being slightly faster. These are functions usually decided upon by the developer and purposely placed there in the linker file prior to compiling the program.

Source: https://www.beningo.com/understanding-the-microcontroller-boot-process/

4.3 Load view and execution view of an image

Image regions are placed in the system memory map at load time. Before you can execute the image, you might have to move some of its regions to their execution addresses and create the ZI output sections. For example, initialized RW data might have to be copied from its load address in ROM to its execution address in RAM.

The memory map of an image has the following distinct views:

Load view	Describes each image region and section in terms of the address where it is located when the image is loaded into memory, that is, the location before image execution starts.
Execution view	Describes each image region and section in terms of the address where it is located during image execution.

The following figure shows these views:

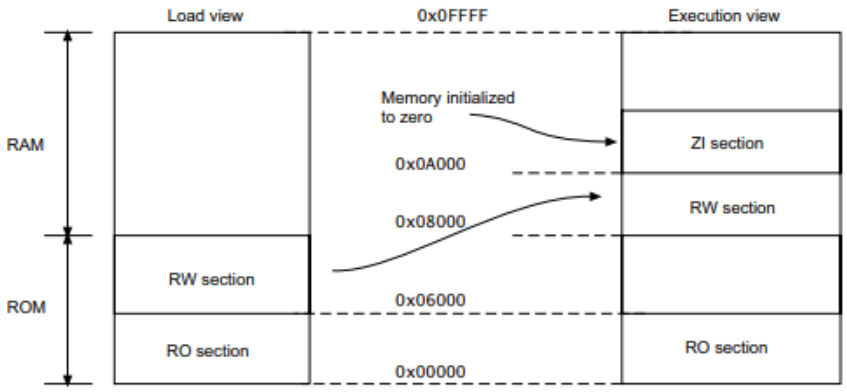
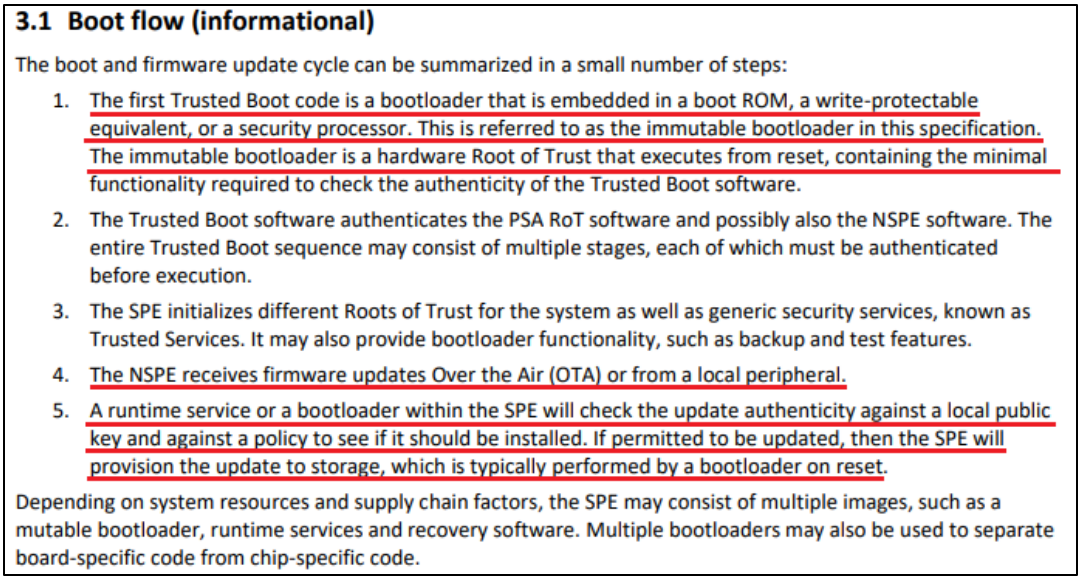


Figure 4-2 Load and execution memory maps

Source: https://documentation-service.arm.com/static/5ea19a939931941038de91a1?token= (Page 32).

58. During initialization and booting of the accused products, the mapped Immutable firmware section ("boot code") is executed first. The Immutable firmware then runs the mapped Updateable firmware ("system code"). Accordingly, the boot code and the system code are orderly executed by the micro-controller to control the booting of the embedded system.



3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

59. The accused products include a connecting interface that is coupled to the micro-controller and to an external data storage device through a data transmission media. The external data storage device stores a second system software.

60. For example, the accused devices support Trusted Base System Architectures like TBSA-M and TBSA-A. TBSA-M specifies the architecture for products that include certain ARM-based micro-controllers. The accused products include two sections of firmware: an Immutable section and an Updateable section. The firmware update of the Updateable section ("second system software") is sent over-the-air via a remote server or by an external local peripheral device ("external data storage device"). The update can be received over-the-air or through an external interface such as USB, UART, SD-MMC, Ethernet, etc. ("connecting interface"). The external interface is capable of being connected to the external data storage device and the micro-controller of the accused devices to facilitate the firmware update process.

This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).

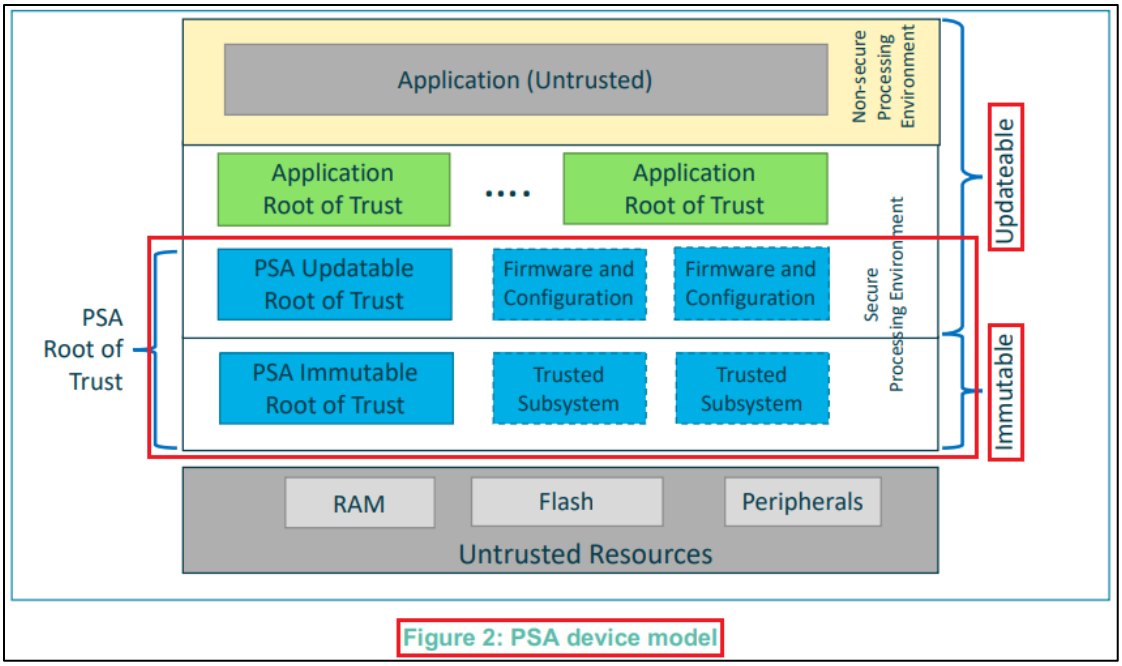
1.2 PSA device model

The PSA Device Model, shown in Figure 2, is a generic reference for defining security properties throughout this document. Designed and deployed devices should map to this reference model.

Figure 2 shows the following domains:

- The PSA Root of Trust (PSA-RoT), which is discussed in section 2, consists of:
  - The PSA Immutable Root of Trust, which is inherently trusted[1] and never changes on a production device.
  - The PSA Updateable Root of Trust, which is trusted only through verification that is anchored to the Immutable PSA Root of Trust. See section 4.
  - Trusted subsystems are, for example, any protected external memory, trusted peripherals, or a secure element, for example a SIM or TPM. Configuration and access control are protected by and attestable by the PSA-RoT.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 4).



Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).

...

3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

2.1.2 Secure firmware update

Significant device firmware should have updateable components which encompass parts of the device RoT all the way up to application software. The location and quantity of deployed TBSA-M devices means that updates should be achievable over a network without requiring physical intervention. This requires the following hardware resources, to ensure that update is performed securely:

- Provision of firmware integrity and authenticity keys.
- Support for approved cryptographic protocols for reception, validation, and installation of new firmware, including monotonic version counters and support for trusted time.
- Provision of non-volatile memory to hold new firmware images and audit logs.
- Resources and mechanisms to remain secure in the event of a failed update, for example, a failsafe backup or a mechanism of removal from Trusted services.

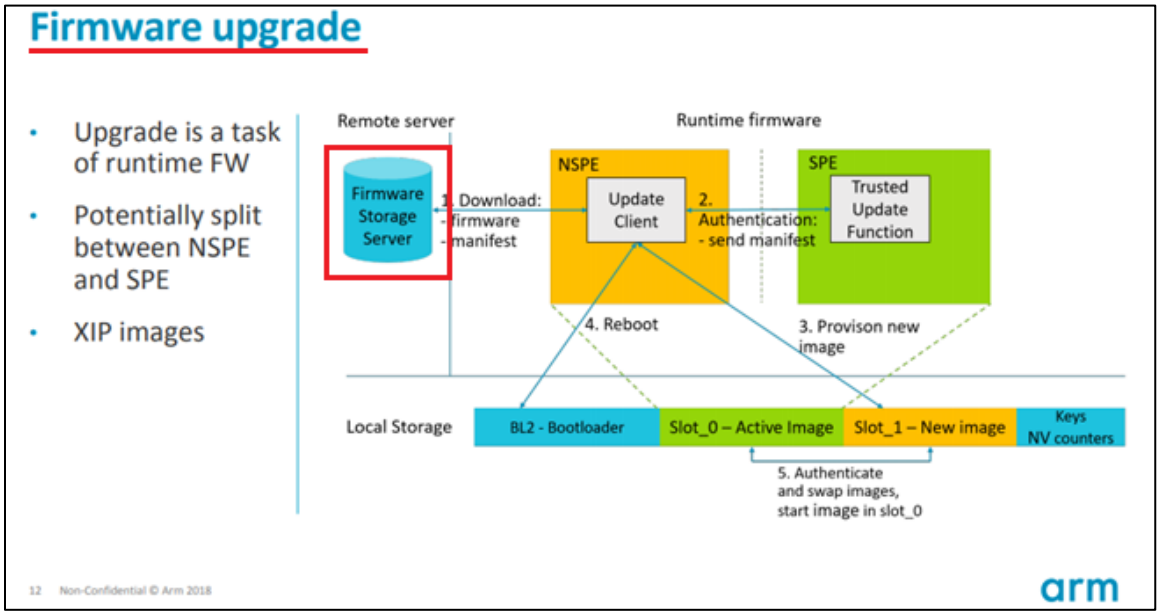
Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 18).

8 Update process (informational)

Some devices require protection against failure of a new image by retention of a known good image, normally the current image. This implies sufficient NVM to store two images. The simplest case is when both images might be stored on the device in eFlash, in which case the eFlash has to be dimensioned for two image slots, a primary slot and a secondary slot. The same principle can be applied for external flash. Following the download and processing of a new image the update client of the device is responsible for programming the new image into the secondary slot and arranging for the device to be rebooted. Images that have been provisioned to storage are known as candidate images.

The update process might fetch images from an external interface such as USB, UART, SD-MMC, NAND, NOR, Ethernet to SoC NVM memories such as NAND Flash, LPPDR2-NVM or any memory as indicated by SoC inputs pins.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 42).

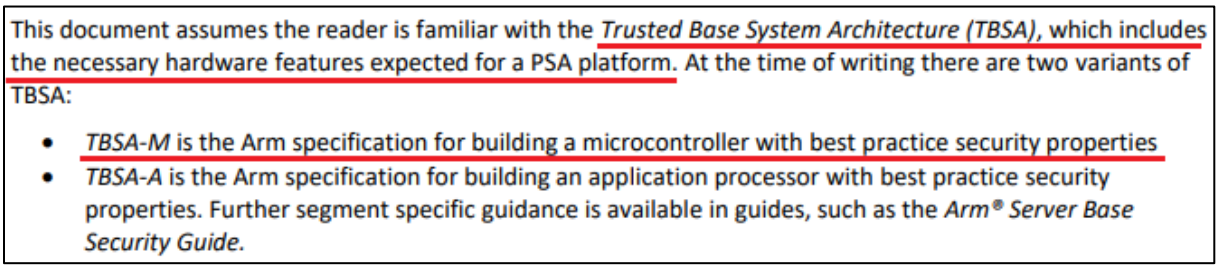


Source: http://connect.linaro.org.s3.amazonaws.com/hkg18/presentations/hkg18-223.pdf (Slide 12).

61. The accused products' boot code includes update agent interface programming (UAIP). The micro-controller is capable of executing the update agent interface programming to read the second system software from the external data storage device through the connecting interface before executing the system code.

62. For example, the accused devices support a secure boot flow process that includes an anti-rollback function. The anti-roll back function ensures that the latest version of the firmware is executed by preventing the execution of older versions.

63. During the secure boot flow process after the accused devices are powered on, the Immutable section of the firmware ("boot code") is executed first. The next section of firmware is executed *only after* the Immutable section of the firmware ("boot code") validates and verifies the version of the Updateable section of the firmware ("system code"). During the boot flow process, the micro-controller's processor executes the "Update Client" and a "Trusted Update Function" (i.e., "update agent interface programming") to retrieve the firmware update ("second system software") from the external server or device ("external data storage device") through a connecting interface, such as USB, UART, SD-MMC, or Ethernet.



This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).

1.2 PSA device model

The PSA Device Model, shown in Figure 2, is a generic reference for defining security properties throughout this document. Designed and deployed devices should map to this reference model.

Figure 2 shows the following domains:

- The PSA Root of Trust (PSA-RoT), which is discussed in section 2, consists of:
  - The PSA Immutable Root of Trust, which is inherently trusted[1] and never changes on a production device.
  - The PSA Updateable Root of Trust, which is trusted only through verification that is anchored to the Immutable PSA Root of Trust. See section 4.
  - Trusted subsystems are, for example, any protected external memory, trusted peripherals, or a secure element, for example a SIM or TPM. Configuration and access control are protected by and attestable by the PSA-RoT.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 4).

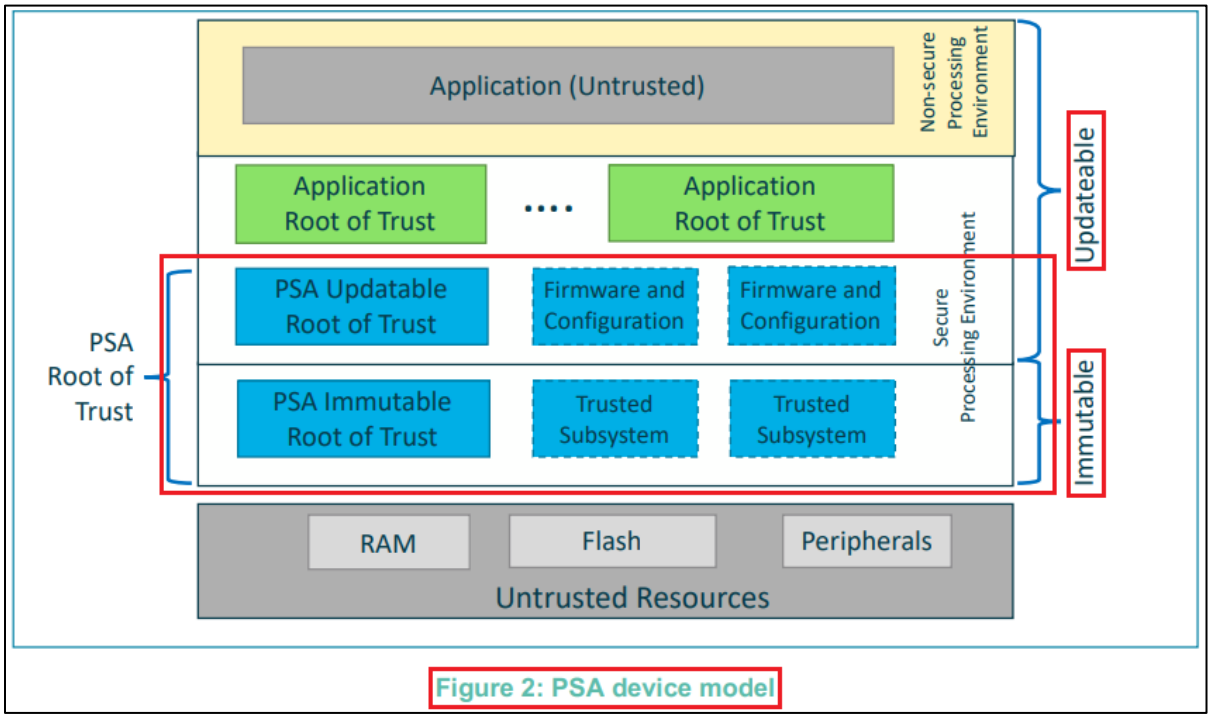


Figure 2: PSA device model

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).

4 PSA-RoT secure boot and firmware update

All PSA devices must support a secure boot flow to ensure only authorized software can be executed on the device. Secure boot, sometimes called verified boot, uses cryptography to verify the next stage code and any metadata. Execution of the next stage proceeds only if any validation checks on the verified metadata pass, for example, version comparison, see section 4.3. In some contexts, the term Measured Boot is used, however, this involves measuring the code, typically for attestation, but without any verification and validity checks.

The secure boot flow must start with an immutable and inherently trusted Boot ROM because it is the trust anchor for the boot validation chain. It is recommended, as shown in Figure 6, that:

- The Boot ROM is small, simple, and verifiable. This minimizes the risk of a vulnerability that cannot be corrected once on the chip, and
- The complex steps are handled by a main bootloader, subject to validity check by the Boot ROM, which can be corrected through a secure firmware update process, see section 4.5.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 14).

4.3 Anti-rollback

Anti-rollback is used to reject earlier versions of the software or data that may contain known errors or vulnerabilities. Secure boot must only allow components that have the same or higher version number than the reference version counter for that component to be executed. To ensure that the component version number is valid, it must be included in the signature of the component and verified before use. The verified version number of each software component must be compared against an on-device reference version number as part of secure boot. To ensure the integrity of the reference version number, it is, typically, stored in an updateable Isolated Location.

The Boot ROM must include an anti-rollback check on all images that it verifies on devices in a secure lifecycle state. Policies for updating the reference counter used by the Boot ROM are covered in section 4.3.1 and section 4.3.2. Subsequent stages in the secure boot chain should include an anti-rollback check on the images that are validated by that stage. The principles that are discussed in section 4.3.1 and section 4.3.2 can be applied.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 16).

3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

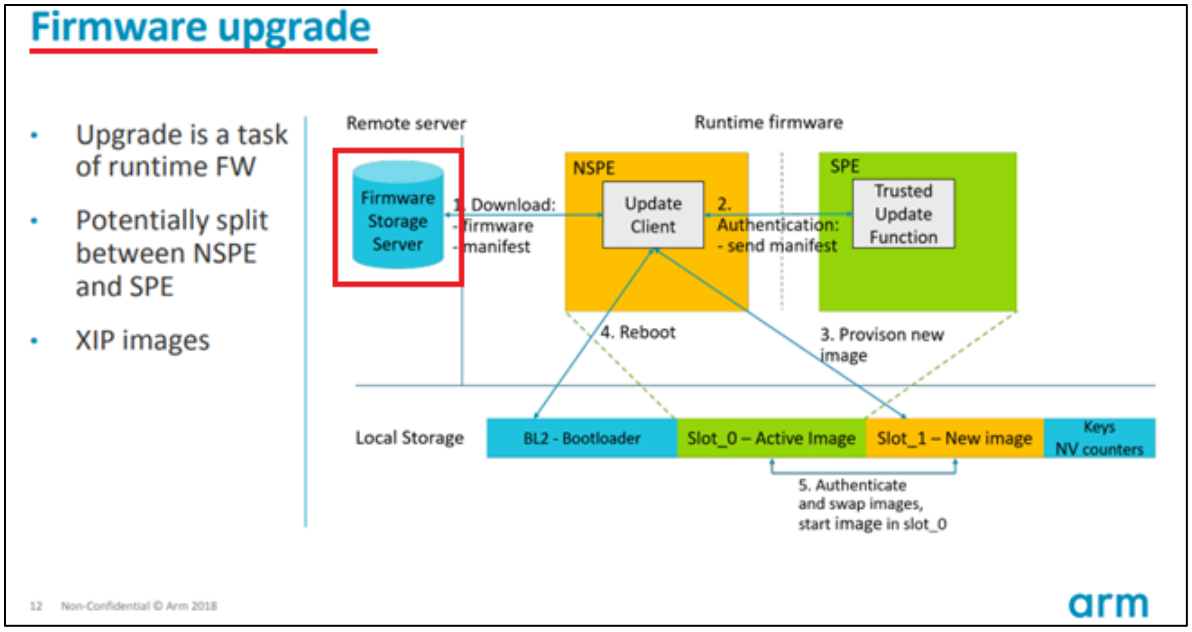
Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

8 Update process (informational)

Some devices require protection against failure of a new image by retention of a known good image, normally the current image. This implies sufficient NVM to store two images. The simplest case is when both images might be stored on the device in eFlash, in which case the eFlash has to be dimensioned for two image slots, a primary slot and a secondary slot. The same principle can be applied for external flash. Following the download and processing of a new image the update client of the device is responsible for programming the new image into the secondary slot and arranging for the device to be rebooted. Images that have been provisioned to storage are known as candidate images.

The update process might fetch images from an external interface such as USB, UART, SD-MMC, NAND, NOR, Ethernet to SoC NVM memories such as NAND Flash, LPPDR2-NVM or any memory as indicated by SoC inputs pins.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 42).



Source: http://connect.linaro.org.s3.amazonaws.com/hkg18/presentations/hkg18-223.pdf (Slide 12).

64. Defendants have had knowledge of the '612 Patent at least as of the date when they were notified of the filing of this action.

65. Liberty Patents has been damaged as a result of the infringing conduct by Defendants alleged above. Thus, Defendants are liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

66. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the '612 Patent.

ADDITIONAL ALLEGATIONS REGARDING INFRINGEMENT AND PERSONAL JURISDICTION

67. Defendants have also indirectly infringed the '612 Patent by inducing others to directly infringe the '612 Patent. Defendants have induced the end-users, Defendants'

customers, to directly infringe (literally and/or under the doctrine of equivalents) the '612 Patent by using the accused products.

68. Defendants took active steps, directly and/or through contractual relationships with others, with the specific intent to cause them to use the accused products in a manner that infringes one or more claims of the patent-in-suit, including, for example, claim 1 of the '612 Patent.

69. Such steps by Defendants included, among other things, advising or directing customers and end-users to use the accused products in an infringing manner; advertising and promoting the use of the accused products in an infringing manner; and/or distributing instructions that guide users to use the accused products in an infringing manner.

70. Defendants performed these steps, which constitute induced infringement, with the knowledge of the '612 Patent and with the knowledge that the induced acts constitute infringement.

71. Defendants were and are aware that the normal and customary use of the accused products by Defendants' customers would infringe the '612 Patent. Defendants' inducement is ongoing.

72. Defendants have also induced their affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on their or their affiliates' behalf, to directly infringe (literally and/or under the doctrine of equivalents) the '612 Patent by importing, selling or offering to sell the accused products.

73. Defendants have a significant role in placing the accused products in the stream of commerce with the expectation and knowledge that they will be purchased by consumers in Texas and elsewhere in the United States.

74. Defendants purposefully direct or control the making of accused products and their shipment to the United States, using established distribution channels, for sale in Texas and elsewhere within the United States.

75. Defendants purposefully direct or control the sale of the accused products into established United States distribution channels, including sales to nationwide retailers. Defendants' established United States distribution channels include one or more United States based affiliates.

76. Defendants purposefully direct or control the sale of the accused products online and in nationwide retailers, including for sale in Texas and elsewhere in the United States, and expect and intend that the accused products will be so sold.

77. Defendants purposefully place the accused products—whether by themselves or through subsidiaries, affiliates, or third parties—into an international supply chain, knowing that the accused products will be sold in the United States, including Texas. Therefore, Defendants also facilitate the sale of the accused products in Texas.

78. Defendants took active steps, directly and/or through contractual relationships with others, with the specific intent to cause such persons to import, sell, or offer to sell the accused products in a manner that infringes one or more claims of the '612 Patent, including, for example, claim 1 of the '612 Patent.

79. Such steps by Defendants included, among other things, making or selling the accused products outside of the United States for importation into or sale in the United States, or knowing that such importation or sale would occur; and directing, facilitating, or influencing their affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons

acting on their or their affiliates' behalf, to import, sell, or offer to sell the accused products in an infringing manner.

80. Defendants performed these steps, which constitute induced infringement, with the knowledge of the '612 Patent and with the knowledge that the induced acts would constitute infringement.

81. Defendants performed such steps in order to profit from the eventual sale of the accused products in the United States.

82. Defendants' inducement is ongoing.

83. Defendants have also indirectly infringed by contributing to the infringement of the '612 Patent. Defendants have contributed to the direct infringement of the '612 Patent by the end-user of the accused products.

84. The accused products have special features that are specially designed to be used in an infringing way and that have no substantial uses other than ones that infringe the '612 Patent, including, for example, claim 1 of the '612 Patent.

85. The special features include, for example, retrieving automatic software updates in an embedded system used in a manner that infringes the '612 Patent.

86. These special features constitute a material part of the invention of one or more of the claims of the '612 Patent and are not staple articles of commerce suitable for substantial non-infringing use.

87. Defendants' contributory infringement is ongoing.

88. Defendants have had actual knowledge of the '612 Patent at least as of the date when they were notified of the filing of this action. Since at least that time, Defendants have

known the scope of the claims of the ’612 Patent, the products that practice the ’612 Patent, and that Liberty Patents is the owner of the ’612 Patent.

89. By the time of trial, Defendants will have known and intended (since receiving such notice) that their continued actions would infringe and actively induce and contribute to the infringement of one or more claims of the ’612 Patent.

90. Furthermore, Defendants have a policy or practice of not reviewing the patents of others (including instructing their employees to not review the patents of others), and thus have been willfully blind of Liberty Patents’ patent rights. *See, e.g.*, M. Lemley, “Ignoring Patents,” 2008 Mich. St. L. Rev. 19 (2008).

91. Defendants’ actions are at least objectively reckless as to the risk of infringing valid patents, and this objective risk was either known or should have been known by Defendants. Defendants have knowledge of the ’612 Patent.

92. Defendants’ customers have infringed the ’612 Patent. Defendants have encouraged their customers’ infringement.

93. Defendants’ direct and indirect infringement of the ’612 Patent has been, and/or continues to be willful, intentional, deliberate, and/or in conscious disregard of Liberty Patents’ rights under the patent-in-suit.

94. Liberty Patents has been damaged as a result of Defendants’ infringing conduct alleged above. Thus, Defendants are liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

JURY DEMAND

Liberty Patents hereby requests a trial by jury on all issues so triable by right.

PRAYER FOR RELIEF

Liberty Patents requests that the Court find in its favor and against Defendants, and that the Court grant Liberty Patents the following relief:

a. Judgment that one or more claims of the ’612 Patent have been infringed, either literally and/or under the doctrine of equivalents, by Defendants and/or all others acting in concert therewith;

b. A permanent injunction enjoining Defendants and their officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents, and all others acting in concert therewith from infringement of the ’612 Patent; or, in the alternative, an award of a reasonable ongoing royalty for future infringement of the ’612 Patent by such entities;

c. Judgment that Defendants account for and pay to Liberty Patents all damages to and costs incurred by Liberty Patents because of Defendants’ infringing activities and other conduct complained of herein, including an award of all increased damages to which Liberty Patents is entitled under 35 U.S.C. § 284;

d. That Liberty Patents be granted pre-judgment and post-judgment interest on the damages caused by Defendants’ infringing activities and other conduct complained of herein;

e. That this Court declare this an exceptional case and award Liberty Patents its reasonable attorney’s fees and costs in accordance with 35 U.S.C. § 285; and

f. That Liberty Patents be granted such other and further relief as the Court may deem just and proper under the circumstances.

Dated: January 27, 2021

Respectfully submitted,

/s/ Zachariah S. Harrington
Matthew J. Antonelli
Texas Bar No. 24068432

matt@ahtlawfirm.com
Zachariah S. Harrington
Texas Bar No. 24057886
zac@ahtlawfirm.com
Larry D. Thompson, Jr.
Texas Bar No. 24051428
larry@ahtlawfirm.com
Christopher Ryan Pinckney
Texas Bar No. 24067819
ryan@ahtlawfirm.com
Rehan M. Safiullah
Texas Bar No. 24066017
rehan@ahtlawfirm.com

ANTONELLI, HARRINGTON
& THOMPSON LLP
4306 Yoakum Blvd., Ste. 450
Houston, TX 77006
(713) 581-3000

Stafford Davis
State Bar No. 24054605
sdavis@stafforddavisfirm.com
Catherine Bartles
Texas Bar No. 24104849
cbartles@stafforddavisfirm.com
THE STAFFORD DAVIS FIRM
815 South Broadway Avenue
Tyler, Texas 75701
(903) 593-7000
(903) 705-7369 fax

Attorneys for Liberty Patents, LLC